מבוא לאלגוריתמים

אלדר פישר

4 באוגוסט 2019

תוכן עניינים

הקדמה קצרה

חוברת זו שמשה במקור עבור הקורס "אלגוריתמים 1" שלימדתי מספר שנים בפקולטה למדעי המחשב בטכניון. חלק מהחומר כאן שואב מההסטוריה של דרכי הלימוד הנהוגות בקורס הוותיק הזה.

רב הסיכויים שכבר התחלתם ללמוד אלגוריתמים עוד מתחילת הלימודים שלכם במדעי המחשב. למשל, סביר שלמדתם את אלגוריתם החיפוש הבינארי כחלק מיסודות התכנות. אפשר להסתכל על החומר כאן כזה שיסכם את "הילדות האלגוריתמית" שלכם. החוברת תשלים את תיבת הכלים הבסיסית בפיתוח וניתוח אלגוריתמים, עם דגש על אלגוריתמים עבור בעיות שניתן למדל בתורת הגרפים (את תיאור הבעיה מעבירים לגרף קומבינטורי, ועליו מפעילים את האלגוריתם). בפרט תלמדו כאן מספר "נכסי צאן ברזל" של האלגוריתמים המוכרים בתחום.

הנחות בסיסיות: כאשר ננתח אלגוריתמים בגרפים, בד"כ נסמן את גרף הקלט ב־G עם קבוצת צמתים V וקבוצת קשתות E, ולרב נניח שהוא פשוט ואינו מכוון. מבחינת האלגוריתמים, נניח לרב שהגרף נתון בצורה של רשימות שכנות (לכל צומת מופיעה רשימת מזהי הצמתים שיש אליהם קשת ממנו), כך שזמן הריצה של האלגוריתם יוכל להיות פונקציה של $|E|$ ו־$|V|$.

דרך הכתיבה: החוברת כתובה בצורה "סיפורית" שמתאימה לקריאה רצופה מההתחלה עד הסוף – מומלץ לפחות לקרוא כל פרק לכשעצמו בצורה כזו. כעיקרון יש דגש על פירוט מלא ככל האפשר של ההוכחות (למעט מקרים ספורים שבהם פירוט כזה היה דורש מיני־קורס שלם), תוך ציון היכן בדיוק משתמשים בהנחות מובלעות. לטענות העזר כאן ניתנים שמות במקום מספרים, על מנת לצמצם את הצורך בדפדוף לאחור כל פעם שמשתמשים בהן.

אלגוריתמי חיפוש בגרפים

הגדרות והאלגוריתם הגנרי

בפרק זה נדון בעיקר בגרפים מכוונים. נתון לנו גרף עם קבוצת צמתים V וקבוצת קשתות E, וכן צומת התחלתי $s \in V$. נרצה לדעת את קבוצת הצמתים $V' \subseteq V$ שעבורם יש מסלולים מ־s. נרצה אף יותר מזה: נרצה לקבל עץ מכוון T עם קבוצת הצמתים V' ושורש s, כך שהקשתות E' שלו נותנות מסלולים מ־s לאיברי V'. דוגמה לבעיה שזהו המודל שלה: נניח שיש לנו כבישים בין ערים (הערים הן צמתים, ויש קשת בין שני צמתים אם יש כביש מתאים בין הערים הנ"ל), ואנחנו מעונינים לדעת לאילו ערים אפשר להגיע מעיר המוצא שלנו. יותר מאוחר, נראה מידול ואלגוריתם גם לשאלה שבה נתונים אורכי הכבישים, ואנחנו מעוניינים בנוסף בדרך הקצרה ביותר מעיר לעיר.

לפני שנמשיך, כדאי להזכיר כאן משפט שאתם מכירים מקומבינטוריקה, אשר ישמש אותנו כמעט בכל פרקי החוברת המזכירים עצים.

משפט (אפיון קומבינטורי של עצים פורשים): כל אחד מהתנאים הבאים עבור תת־גרף $G' = (V, E')$ של $G = (V, E)$ שקול לכך ש־G' הוא עץ פורש לא מכוון של G.

- הגרף G' קשיר וחסר מעגלים.
- הגרף G' קשיר ומתקיים $|E| = |V| - 1$.

- הגרף G' חסר מעגלים ומתקיים $|E| = |V| - 1$.

אפשר להשתמש במשפט הזה גם עבור עצים מכוונים, כי ההבדל בין עץ מכוון לעץ לא־מכוון הוא הגדרה של אחד הצמתים בו כשורש, כשכל הקשתות יהיו מכוונות ממנו והלאה (בעץ מכוון, קשת מהצורה uv תקיים ש־u על המסלול בעץ משורש העץ ל־v).

עתה נראה "אלגוריתם גנרי" לביצוע המשימה של מציאת עץ מסלולים כפי שהוגדר למעלה. אפשר לקרוא לזה גם "שלד–אלגוריתם", כי יש בו חלק "לא דטרמיניסטי" שבו בחירת הצעד הבא מוגדרת חלקית בלבד, ויש יותר מאפשרות אחת. היתרון בניתוח אלגוריתם כזה הוא ביכולת לתת הבטחות לגבי מספר אלגוריתמים מוגדרים היטב, לאחר שמראים שהם מהווים מקרים פרטיים שלו. במהלך האלגוריתם אנחנו גם נתחזק מצביע $p(v)$ מכל $v \in V'$ לצומת האב בעץ המכוון. הרעיון הכללי של האלגוריתם יהיה להתחיל מ־$V' = \{s\}$, ואז להרחיב את הקבוצה לפי "חבר מביא חבר".

אלגוריתם גנרי לחיפוש משורש s

קלט: גרף (בד"כ מכוון) $G = (V, E)$, צומת התחלה $s \in V$

פלט: עץ מכוון $T = (V', E')$ עם שורש s, פונקצית מצביע לאב p

- **אתחול:** $V' \leftarrow \{s\},\ E' \leftarrow \emptyset$ ולכל $v \in V$ מציבים $p(v) \leftarrow \mathrm{nil}$
- **כל עוד** יש קשת vw עם $v \in V'$ ו־$w \in V \setminus V'$
  - מוסיפים את w ל־V' ואת vw ל־E'
  - מציבים $p(w) \leftarrow v$

האלגוריתם "גנרי" כי לא ציינו איזו קשת vw אנחנו בוחרים בכל שלב ממבחר הקשתות המקיימות את התנאי. ננתח עתה את זמן הריצה המקסימלי. בכל שלב בלולאה אנחנו מוסיפים צומת ל־V', ולכן אין יותר מ־$O(|V|)$ שלבים. הסריקה בכל שלב במקרה הכי גרוע לוקחת זמן $O(|E|)$, ולכן זמן הריצה המקסימלי הוא $O(|V||E|)$. הזמן $O(|V|)$ של שלב האתחול נבלע בזמן הכולל. יותר מאוחר נראה אלגוריתמים שמממשים את האלגוריתם הגנרי, אבל רצים בזמן יותר קצר.

עתה נראה נכונות, בלי קשר לאיך אנחנו בוחרים את הקשת בכל שלב. ראשית נראה שלא נכנסים ל־V' צמתים שאין אליהם מסלול מ־s: ההוכחה היא באינדוקציה על מספר השלבים שבוצעו. אם עד השלב ה־k נכנסו רק צמתים שיש אליהם מסלול, ועכשיו הכנסנו צומת w לפי הקשת vw, הרי שאפשר לקחת את המסלול מ־s ל־v ולהוסיף אליו את הקשת vw כדי לקבל מסלול מ־s ל־w. זה גם מראה לנו שיטה למציאת המסלולים מתוך פלט האלגוריתם, ע"י הפעלות חוזרות של p החל מ־w עד שמגיעים ל־s.

מצד שני, נראה שעבור הצמתים שלא נכנסו ל־V' אין מסלול מ־s: האלגוריתם עוצר רק כאשר אין קשתות כלל מ־V' ל־$V \setminus V'$ (או כאשר $V = V'$). לפי מה שאתם יודעים בקומבינטוריקה, הדבר אומר שאין מסלול מאף צומת ב־V', כולל s, לאף צומת ב־$V \setminus V'$ (תזכורת מהירה: אם היה מסלול $u_1, \ldots, u_k$

מ־V' ל־$V \setminus V'$, אז היה אפשר לבחור את ה־i המקסימלי עבורו $u_i \in V'$, ואז u_iu_{i+1} היתה קשת מ־V' ל־$V \setminus V'$, בסתירה).

עבור ההוכחה ש־E' היא קבוצת קשתות של עץ, נשים לב ש־$E' = \{(p(u), u) : u \in V' \setminus \{s\}\}$. אפשר מכאן להוכיח שזהו עץ במספר שיטות. למשל, הקשירות מעל V' נובעת ממה שהראינו למעלה על מסלולים מ־s לצמתים ב־V', וקל לראות שמתקיים $|E'| = |V'| - 1$.

אפשר לסכם את הכל במשפט הפורמלי הבא.

משפט (נכונות האלגוריתם הגנרי): האלגוריתם הגנרי תמיד ייתן תת־גרף (V', E') שהוא עץ, וקבוצת צמתיו היא קבוצת כל הצמתים הנגישים מ־s.

לפני המעבר לאלגוריתמים ספציפים, נדון באפשרות "לכסות" את כל צמתי הגרף כאשר אין לנו צומת התחלה ספציפי s. במקרה כזה נוסיף עצי חיפוש זרים כאשר כל פעם נתחיל מצומת שלא ביקרנו בו תוך כדי חיפוש מצומת קודם. האלגוריתם המלא יראה כך.

אלגוריתם גנרי לחיפוש מכסה־גרף

קלט: גרף $G = (V, E)$

פלט: יער מכוון $F = (V', E')$, פונקצית מצביע לאב p

- **אתחול:** $V' \leftarrow \emptyset,\ E' \leftarrow \emptyset$ ולכל $v \in V$ מציבים $p(v) \leftarrow \mathrm{nil}$
- **כל עוד** קיים צומת $v \in V \setminus V'$
  - מוסיפים את v ל־V'
  - **מבצעים** את הלולאה המרכזית של אלגוריתם החיפוש הגנרי, ללא שלב האתחול שלו

עבור גרף לא מכוון, אלגוריתם זה יחזיר יער פורש המכיל עץ פורש לכל רכיב קשירות. במקרה של גרף מכוון, בשילוב אלגוריתם החיפוש לעומק שנלמד בהמשך, יהיה ניתן למצוא עם פרוצדורה כזו את הפירוק של הגרף לרכיבים קשירים־היטב.

חיפוש לרוחב

חיפוש לרוחב, BFS – Breadth First Search, הוא מקרה פרטי (נקרא לזה "יישום") של האלגוריתם הגנרי, שבו בכל שלב אנחנו מנסים את הקשתות היוצאות מכל הצמתים הנוכחיים ב־V' "בבת אחת". למעשה יש לנו שכבות, כאשר השכבה ה־i היא השכבה של כל הצמתים שהיו מחוברים לצמתים שנכנסו לשכבה ה־$i-1$, כשהשכבה ה־0 מכילה רק את s.

חיפוש לרוחב Breadth First Search

קלט: גרף (אפשר מכוון) $G=(V,E)$, צומת התחלה $s\in V$

פלט: עץ מכוון $T=(V',E')$ עם שורש s, פונקצית מצביע לאב p, פונקצית עומק d

- **אתחול:** $V'\leftarrow\{s\},\ E'\leftarrow\emptyset,\ i\leftarrow 0$, ולכל $v\in V$ מציבים $p(v)\leftarrow\mathrm{nil}$; כמו כן $d(s)\leftarrow 0$, ולכל $v\in V\setminus\{s\}$ מציבים $d(v)\leftarrow+\infty$
- **כל עוד** קיימים ב־V' צמתים v עבורם $d(v)=i$
  - **עבור כל** קשת vw עם $d(v)=i$ ו־$w\in V\setminus V'$
    * מוסיפים את w ל־V' ואת vw ל־E'
    * מציבים $p(w)\leftarrow v$ ו־$d(w)\leftarrow i+1$
  - מציבים $i\leftarrow i+1$

ראשית צריך להראות שזהו באמת מקרה פרטי של האלגוריתם הגנרי. הדבר הראשון שצריך לשים לב אליו, הוא שכאשר אנחנו מתחילים לעבור על הצמתים עם $d(v)=i$, לא יהיו קשתות vw בין V' ל־$V\setminus V'$ עבורן $d(v)<i$. זאת מכיוון שהגדלנו את i רק אחרי שטיפלנו בכל הקשתות עבורן $d(v)=i$, ולא יווצרו לנו קשתות חדשות כאלו (כל הצמתים שנוספים אח"כ ל־V' יהיו עם ערכי d גדולים יותר). כמו כן לא יהיו קשתות vw בין V' ל־$V\setminus V'$ שעבורן $d(v)>i+1$, פשוט כי לא הצבנו ערכי d כאלו עד תחילת המעבר, והקשתות עבורן $d(v)=i+1$ יטופלו בפעם הבאה שמגדילים את i. אנחנו עוצרים רק לאחר ערך של i שעבורו לא מצאנו קשתות vw עם $d(v)=i$ ו־$w\in V\setminus V'$ (ואז לאחר ההגדלה של i לא יהיו צמתים עם ערך d מתאים והלולאה תיעצר), ולפי הדיון למעלה זה קורה רק לאחר שאין קשתות מ־V' ל־$V\setminus V'$.

לפני שנמשיך, כדאי לראות לשים לב שיש שיטה שקולה לכתוב את האלגוריתם לחיפוש לרוחב, באמצעות "הרצה עם תוספות" של האלגוריתם הגנרי.

תוספות לאלגוריתם הגנרי שהופכות אותו לחיפוש לרוחב

- בעת האתחול, מציבים גם $d(s)\leftarrow 0$, ולכל $v\in V\setminus\{s\}$ מציבים $d(v)\leftarrow\infty$
- כאשר יש מספר קשתות vw עם $v\in V'$ ו־$w\in V\setminus V'$, בוחרים אחת עם $d(v)$ מינימלי
  - כאשר מטפלים ב־vw כזו, מציבים בנוסף $d(w)\leftarrow d(v)+1$

אפשר לראות (הדבר מושאר לכם כתרגיל) שבגרסה הזו, המינימום על כל ערכי $d(v)$ עבור $v\in V'$ שיש להם לפחות קשת אחת לצומת ב־$V\setminus V'$ משחק את התפקיד של i.

עתה נראה את היתרון באלגוריתם החיפוש לרוחב. עבור שני צמתים u,v כל שהם, נסמן ב־$\mathrm{dist}(u,v)$ את אורך המסלול הקצר ביותר ביניהם.

משפט: לכל $v \in V'$ יתקיים $d(v) = \mathrm{dist}(s,v)$.

הוכחה: ראשית מראים באינדוקציה שלכל $v \in V'$ יש מסלול כל שהוא באורך $d(v)$ מ־s אליו, ומכאן שמתקיים $d(v) \geq \mathrm{dist}(s,v)$. הבסיס הוא שרק $v = s$ מקיים $d(v) = 0$. לצומת v עבורו $d(v) = k$, נשים לב שמתקיים $d(p(v)) = k - 1$. נחבר אז את המסלול מאורך $k-1$ מ־s ל־$p(v)$ (קיים מסלול כזה לפי הנחת האינדוקציה) עם הקשת $v, p(v)$, ונקבל את המבוקש.

בכיוון השני, מראים באינדוקציה שאם יש מ־s ל־v מסלול מאורך k אז מתקיים $d(v) \leq k$. הבסיס הוא $k = 0$, שמתקיים רק אם $v = s$. עתה, אם הטענה נכונה עבור $k-1$, אז לוקחים את u להיות הצומת לפני v על המסלול מאורך k מ־s ל־v. לפי הנחת האינדוקציה מתקיים $d(u) \leq k-1$, ואז לכל המאוחר כאשר $i = k-1$ הצומת v יכנס ל־V' ויקבל שם ערך $d(v) = i+1 \leq k$. ■

שימו לב שהוכחנו גם שהעץ המתקבל מהרצת האלגוריתם הוא עץ מסלולים קצרים ביותר – עבור כל $v \in V'$, אם לוקחים את הסידרה $v, p(v), p(p(v)), \ldots, s$ והופכים אותה (מתחילים ב־s ומסיימים ב־v) מקבלים מסלול ב־T מאורך $\mathrm{dist}(s,v)$: מ־$d(p(v)) = d(v) - 1$ כאשר $d(s) = 0$ נובע שאורך המסלול הוא $d(v)$, והוכחנו $d(v) = \mathrm{dist}(s,v)$, ולכן זהו מסלול קצר ביותר בין שני הצמתים.

עד עתה לא ניסינו לשפר את זמן הריצה מעבר ל־$O(|V||E|)$. התכונה שבכל פעם עוברים על קשתות משכבה בודדת שאח"כ לא חוזרים אליה תאפשר לנו לממש את האלגוריתם בצורה ש"נקדיש תשומת לב" לכל קשת רק פעמיים במהלך האלגוריתם (פעם אחת עבור כל צומת של הקשת), וכך נגיע לסיבוכיות זמן משופרת של $O(|V| + |E|)$. האלגוריתם הבא הוא מימוש של חיפוש לרוחב באמצעות מבנה נתונים של תור (Queue).

חיפוש לרוחב – מימוש באמצעות תור

קלט: גרף (אפשר מכוון) $G = (V, E)$, צומת התחלה $s \in V$

פלט: עץ מכוון $T = (V', E')$ עם שורש s, פונקצית מצביע לאב p, פונקצית עומק d

- **אתחול:** $p(s) \leftarrow \mathrm{nil}$, $d(s) \leftarrow 0$, $E' \leftarrow \emptyset$, $V' \leftarrow \{s\}$, וכן $p(v) \leftarrow \mathrm{nil}$, $d(v) \leftarrow \infty$ לכל $v \in V \setminus \{s\}$.
- מאתחלים את התור Q להכיל את הצומת s בלבד. בסימון מתמטי: $Q \leftarrow \langle s \rangle$
- **כל עוד** Q איננו ריק
  - מוציאים את הצומת הראשון v מהתור
  - **עבור כל** קשת מהצורה $vw \in E$, **אם** $w \in V \setminus V'$
    - מוסיפים את w ל־V' ואת vw ל־E'
    - מציבים $p(w) \leftarrow v$, $d(w) \leftarrow d(v) + 1$
    - מוסיפים את w לסוף התור Q

על מנת לנתח את זמן הריצה, נשים לב שכל צומת יכול להיכנס לתור לא יותר מפעם אחת, מכיוון שרק צמתים שאינם ב־V' מוכנסים לתור (ובמקביל גם ל־V'). זמן הטיפול בצומת v הוא $O(1)$ כשהוא נכנס לתור ו־$O(\deg(v))$ כשהוא יוצא מהתור, כך שלאחר הסיכום על כל הצמתים יוצא זמן ריצה של $O(\sum_{v\in V}(1+\deg(v)))=O(|V|+|E|)$. זמן האתחול $O(|V|)$ נבלע בזה (את V' אפשר למשל לייצג ע"י מערך של ביטים).

צריך אבל גם להראות שזהו מימוש של BFS. ראשית, מוכיחים באינדוקציה על הכנסות והוצאות מהתור Q את השמורה הבאה: בכל שלב נתון, סידרת הערכים $\langle d(w) : w \in Q\rangle$ היא סידרה מונוטונית לא־יורדת, ובנוסף ההפרש בין האיבר הראשון לאחרון בה הוא לא יותר מ־1. מכך נובע גם שסידרת ערכי d של הצמתים לפי סדר כניסתם לתור (או יציאתם ממנו, שזה אותו סדר) היא מונוטונית לא־יורדת, ו"רציפה" (ההפרש בין שני איברים עוקבים אינו עולה על 1). בעת האתחול (כשהתור עם איבר בודד) השמורה נכונה. להוכחה שהיא נשמרת לכל אורך האלגוריתם, שימו לב שכאשר הוצאנו צומת v מראש התור שעבורו $d(v)=i$, אז מההנחה שהשמורה התקיימה עד עכשיו, שאר הצמתים בתור הם בעלי ערכי $d(u)$ של i או $i+1$. אנחנו הוספנו לסוף התור אך ורק צמתים עם ערך של $i+1$, ולכן לא שינינו מצב זה, וגם לא הכנסנו צומת עם ערך קטן מהערך של הצומת האחרון שהיה שם קודם.

מהסתכלות זו נובע שלאחר הצעד של הטיפול בצומת u האחרון שעבורו $d(u)=k-1$, כל הצמתים $w\in Q$ יקיימו עתה $d(w)=k$. כמו כן אלו יהיו הצמתים היחידים ב־V' עבורם מתקיים השוויון הנ"ל, ולא יהיו ב־V' בשלב זה צמתים עם ערכי d גבוהים יותר. מכיוון שהאלגוריתם יטפל בצמתים אלו לפני שיטפל בכל צומת עתידי אחר (שיכול להיכנס לתור רק אחריהם), אנחנו רואים שמתקבל כאן ביצוע של אלגוריתם החיפוש לרוחב עבור הערך $i=k$. מהדיון למעלה, בסופו של דבר האלגוריתם מתבצע עבור כל ערכי i לפי סידרם החל מ־$i=0$ (שזהו המצב ההתחלתי $Q=\langle s\rangle$).

חיפוש לעומק

יש גישה מרכזית שניה לקביעת סדר החיפוש באלגוריתם הגנרי, והיא חיפוש לעומק DFS – Depth First Search. כאן, כל פעם שמגלים צומת חדש להכניס ל־V', בודקים את כל "ההמשכים האפשריים" של החיפוש דרכו לפני ש"חוזרים" ממנו וממשיכים לצמתים שהמסלול אליהם לא עובר דרכו.

חישבו על הפרוצדורה הרקורסיבית הבאה, אשר מקבלת מזהה צומת $v\in V$ כקלט:

פרוצדורת חיפוש רקורסיבית $\mathrm{Search}(v)$

קלט: גרף (אפשר מכוון) $G=(V,E)$, צומת חיפוש נוכחי $v\in V$

משתנים גלובלים: עץ (או יער) מכוון $T=(V',E')$ עם $V'\subseteq V$ ו־$E'\subseteq E$, פונקצית מצביע לאב p

- **לכל** צומת w שהוא שכן של v
  - **אם** $w\in V\setminus V'$
    - מכניסים את w ל־V', את vw ל־E', ומציבים $p(w)\leftarrow v$

* **קוראים באופן רקורסיבי** לפרוצדורה $\text{Search}(w)$ שתמשיך לחפש מ־w

---

האלגוריתם המלא יכלול איתחול וקריאה ראשית לפרוצדורה:

חיפוש לעומק Depth First Search

קלט: גרף (אפשר מכוון) $G = (V, E)$, צומת התחלה $s \in V$

פלט: עץ מכוון $T = (V', E')$ עם שורש s, פונקצית מצביע לאב p

- **אתחול:** $V' \leftarrow \{s\}$, $E' \leftarrow \emptyset$ ולכל $v \in V$ מציבים $p(v) \leftarrow \text{nil}$
- **קוראים** לפרוצדורה $\text{Search}(s)$ שתתחיל לחפש מ־s

לפני שנמשיך, נראה שזה אכן מיישם את האלגוריתם הגנרי. לשם כך נשים לב שהפרוצדורה הרקורסיבית, אחרי שנקראה עבור צומת v, תחזור רק לאחר שווידאה שכל השכנים שלו נמצאים ב־V' (את אלו שאינם בו הפרוצדורה תכניס בעצמה). כמו כן, כל הכנסה של צומת ל־V' מלווה מיידית בקריאה לפרוצדורת החיפוש עבורו. מכאן שהתכנית תסתיים (לאחר סיום כל הקריאות הרקורסיביות לפרוצדורה) רק לאחר שאין צמתים ב־V' עם שכנים ב־$V \setminus V'$.

בקשר לסיבוכיות הזמן, נשים לב שהפרוצדורה הרקורסיבית, בכל מהלך ריצת האלגוריתם, לא תיקרא יותר מפעם אחת לכל $v \in V$. זאת מכיוון שכל קריאה כזו מלווה בהכנסה של v ל־V', אשר שמור באופן גלובלי (לא מקומי), ואין קריאות כאלו עבור צמתים שכבר נמצאים ב־V'.

בתוך קריאה בודדת לפרוצדורה עם צומת v, נבדוק את זמן הריצה שלה, לא כולל את זמן הריצה של קריאות רקורסיביות שהיא עושה (מכיוון שאח"כ סוכמים את הזמנים לכל הקריאות בלי קשר להיכן נעשו, הסכום יכלול גם את הזמן של הקריאות הרקורסיביות). זמן הריצה הנ"ל הוא $O(1 + \deg(v))$. מכאן מקבלים זמן ריצה כולל של $O(\sum_{v \in V}(1 + \deg(v)) = O(|V| + |E|)$. זמן האתחול $O(|V|)$ נבלע בביטוי הזה.

במקום קריאות רקורסיביות, אפשר לתחזק באופן "ידני" מחסנית עם הצמתים שנבדקים באותו רגע. המימוש נעשה בצורה הבאה. יש כאן גם תחזוק של "זמן גילוי" עבור כל צומת, שנשתמש בו בהמשך. פרט לתחזוק של זמן הגילוי $k(v)$, אין בגרסה הזו של האלגוריתם דבר פרט לסימולציה של האלגוריתם הרקורסיבי באמצעות המחסנית.

חיפוש לעומק – ממומש ע"י מחסנית

קלט: גרף (אפשר מכוון) $G = (V, E)$, צומת התחלה $s \in V$

פלט: עץ מכוון $T = (V', E')$ עם שורש s, פונקצית מצביע לאב p, פונקצית זמן גילוי k

- **אתחול:**
  - מציבים $i \leftarrow 1$, $E' \leftarrow \emptyset$, $V' \leftarrow \{s\}$, ולכל $v \in V$ מציבים $k(v) \leftarrow +\infty$, $p(v) \leftarrow \text{nil}$
  - מאתחלים את המחסנית S להחזיק את הצומת s בלבד, ז"א $S \leftarrow \langle s \rangle$
  - מציבים $k(s) \leftarrow 1$ ו־$j(s) \leftarrow 0$ (ערכי j חשובים רק עבור צמתים שנמצאים באותו רגע במחסנית, ולכן אפשר לאחסן אותם במחסנית לצד מזהי הצמתים, במקום להחזיק מערך נפרד שלהם)
- **כל עוד** המחסנית לא ריקה
  - יהי v הצומת בראש המחסנית
  - מוסיפים 1 ל־$j(v)$
  - **אם** $j(v) > \deg(v)$ אז
    * מוציאים את v מהמחסנית
  - **אחרת, אם** השכן ה־$j(v)$ של v נמצא ב־V'
    * לא עושים כלום (אבל בהמשך נראה אלגוריתם עם פעולה במקרה הזה)
  - **אחרת, אם** השכן ה־$j(v)$ של v אינו ב־V'
    * יהי w השכן הנ"ל
    * מוסיפים 1 ל־i
    * מוסיפים את w ל־V', את vw ל־E', ומציבים $k(w) \leftarrow i$, $p(w) \leftarrow v$
    * מכניסים את w לראש המחסנית ומציבים $j(w) \leftarrow 0$

לפני שנמשיך, כדאי לנסח מספר טענות נוספות לגבי אלגוריתם החיפוש לעומק ועץ החיפוש המתקבל ממנו (אנחנו יודעים שזה עץ כי הראנו שזה יישום של האלגוריתם הגנרי).

טענה (המחסנית מכילה מסלול): בכל רגע נתון המחסנית $S = \langle u_0, u_1, \ldots, u_{r-1}, u_r \rangle$, כאשר u_r הוא ראש המחסנית ו־$u_0 = s$ הצומת העמוק ביותר בה, מתאימה למסלול מ־u_0 ל־u_r בעץ החיפוש המכוון. בפרט $p(u_j) = u_{j-1}$ לכל j בין 1 ל־r.

הוכחה: הדבר נובע מאינדוקציה על מספר פעולות ההכנסה וההוצאה במחסנית מאז תחילת האלגוריתם (בכל פעם שמכניסים צומת u_r ל־S, מציבים את $p(u_r)$ להיות u_{r-1}), כאשר הבסיס הוא כשהמחסנית אותחלה להכיל רק את הצומת s. ■

טענה (קשר בין יחס צאצאות למחסנית): עבור $u, v \in V'$ כל שהם, v הוא צאצא של u בעץ החיפוש אם ורק אם u היה במחסנית כאשר v הוכנס לתוכו.

הוכחה: הכיוון של "אם" נובע ישירות מהטענה הקודמת. עבור הכיוון של "רק אם", נשים לב שנובע מהטענה הקודמת שמצב המחסנית בעת הכנסת הצומת $u_r = v$ הוא המסלול המלא מ־v לשורש. אם u לא היה שם בעת ההכנסה, אז הוא לא על המסלול הנ"ל ולכן אינו יכול להיות אב קדמון של v. ■

הלמה הבאה מגלמת את ה"אופי" היחודי של חיפוש לעומק.

למה (ראיה קדימה): אם כאשר u הוכנס למחסנית היה מסלול ממנו לצומת v שכל צמתיו עוד לא התגלו (לא היו ב־V' בעת ההכנסה של u), אז v יהיה צאצא של u בעץ החיפוש המתקבל לבסוף.

הוכחה: באינדוקציה על אורך המסלול. הבסיס הוא כאשר $u = v$. אחרת נסמן את המסלול ב־$u = v_0, v_1, \dots v_r = v$. באיזה שהוא שלב v_1 יכנס למחסנית לפני ההוצאה של u, כי לא מוציאים את u לפני שבודקים את כל שכניו. יכול להיות אבל שהוא לא יהיה הצומת הראשון מהמסלול שיכנס. נניח ש־v_j הוא הצומת הראשון מהמסלול שנכנס למחסנית לפני ההוצאה של u. בשלב זה v_j הופך לצאצא של u לפי הטענה שהמחסנית מכילה מסלול. כמו כן, v יהיה צאצא של v_j מהנחת האינדוקציה על $v_j, v_{j+1}, \dots, v_r = v$. משני אלו נובע ש־$v$ יהיה צאצא של u. ■

הלמה על ראיה קדימה נקראת לפעמים בספרות "למת המסלול הלבן" (צומת "לבן" הוא צומת שעוד לא נתגלה והוכנס ל־V').

לפני שנמשיך, כדאי להשוות מה יוצא מחיפוש לרוחב (BFS) מול חיפוש לעומק (DFS) כאשר מפעילים אותם על קליק (גרף לא מכוון בעל כל הקשתות האפשריות): חיפוש לרוחב יתן עץ חיפוש שהוא כוכב, בעוד חיפוש לעומק יתן עץ חיפוש המורכב ממסלול אחד דרך כל הצמתים.

כדאי גם לדעת שכמו שהיה לחיפוש לרוחב ניסוח שקול שבו הקשת שמטפלים בה נבחרת לפי צומת מוצא עם ערך מרחק d מינימלי שם, אפשר היה לתת לחיפוש לעומק ניסוח שקול (לא כולל תיאור מימוש), שבו הקשת שמטפלים בה נבחרת לפי צומת מוצא ב־V' עם זמן גילוי k מקסימלי. אתם מוזמנים לנסות להוכיח שהקריטריון הזה אכן מתאים לחיפוש לעומק, ע"י הוכחת השמורות המתאימות על המחסנית (כעיקרון צריך להראות שהצומת בראש המחסנית הוא בדיוק הצומת ב־V' עם זמן הגילוי המקסימלי שעוד נותרו לו שכנים מחוץ ל־V').

הגרסה של כיסוי גרף (במקום חיפוש מצומת נתון s) חשובה במיוחד עבור חיפוש לעומק. נציג אותה כאן. נעיר ששתי הלמות האחרונות למעלה ניתנות להכללה גם לגרסה זו.

אלגוריתם חיפוש לעומק מכסה־גרף

קלט: גרף $G = (V, E)$

פלט: יער מכוון $F = (V', E')$, פונקצית מצביע לאב p, פונקצית זמן גילוי k

- **אתחול:** $V' \leftarrow \emptyset,\ E' \leftarrow \emptyset,\ i \leftarrow 0$ ולכל $v \in V$ מציבים $p(v) \leftarrow \text{nil},\ k(v) \leftarrow +\infty$
- **כל עוד** קיים צומת $v \in V \setminus V'$
  - מגדילים את i ב־1
  - מוסיפים את v ל־V' ומציבים $k(v) \leftarrow i$
  - **מאתחלים** $j(v) \leftarrow 0$ ואת המחסנית S להחזיק את v בלבד, **ומבצעים** את הלולאה המרכזית של אלגוריתם החיפוש לעומק (בלי האתחולים שם) עד לריקון המחסנית

לסיום הדיון הכללי, נחשוב על הקיטלוג הבא של קשתות בגרף המושרה על עץ חיפוש כל שהוא: יש את קשתות העץ עצמו, "קשתות לאחור" מצומת אל אב קדמון (גם לולאות יחשבו לקשתות לאחור), "קשתות לפנים" מצומת אל צאצא (בגרף לא מכוון אין הבדל בין קשתות לפנים וקשתות לאחור – נהוג אז לקרוא לכל הקשתות האלו קשתות לאחור), ו"קשתות חוצות" שאינן אף אחת מהסוגים הנ"ל (מיד נראה שעבור גרף לא מכוון לא יהיו קשתות כאלו בעץ המתקבל מחיפוש לעומק). כאשר משתמשים בגרסת האלגוריתם אשר מכסה את כל הצמתים, אז גם קשת בין שני עצי חיפוש שונים ביער המתקבל תיקרא קשת חוצה.

מהלמה על ראיה קדימה נובעת הטענה הבאה, שגם אותה לא קשה להכליל לגרסת האלגוריתם המכסה את כל צמתי הגרף.

טענה (אין קשתות חוצות): בעץ שנוצר מחיפוש לעומק בגרף לא מכוון אין קשתות חוצות.

הוכחה: נניח ש־uv היא קשת חוצה, ונניח גם שמתקיים $k(u) < k(v)$ (כאן משתמשים בזה שהגרף אינו מכוון, ולכן אפשר אם צריך להחליף את u ו־v). בעת ביצוע החיפוש לעומק, ברגע ש־u הוכנס למחסנית היה מסלול (של קשת אחת) ממנו ל־v שלא עובר דרך צמתים שכבר התגלו, ולכן לפי הלמה על ראיה קדימה v היה צריך להיות צאצא של u, בסתירה. ■

הטענה למעלה אינה נכונה עבור גרפים מכוונים. קחו לדוגמה את הגרף על $\{s,u,v\}$ עם הקשתות $\{su,sv,uv\}$. יכול להתקבל עבורו עץ חיפוש שבו שני הצמתים u,v הם בנים של השורש s (אם v הוכנס ל־V' לפני u).

אם חוזרים לגרפים לא מכוונים, אז נהוג לקרוא "עץ חיפוש לעומק" לכל עץ מכוון אשר פורש רכיב קשירות של הגרף ושעבורו אין לגרף קשתות חוצות. אתם מוזמנים להראות שכל עץ כזה אכן יכול להתקבל מביצוע חיפוש לעומק, לאחר סידור מתאים של רשימות השכנויות של הצמתים אשר מסופקות לאלגוריתם.

האלגוריתם של Tarjan למציאת רכיבים קשירים היטב

הערה חשובה: האלגוריתם שנמצא במרבית הספרות עבור רכיבים קשירים היטב הוא זה של Kosaraju, השונה מהותית מהאלגוריתם שמוצג כאן. האלגוריתם כאן דומה ב"רוחו" לאלגוריתם (היותר קשה) שמוצג בהמשך למציאת רכיבים אי־פריקים בגרף לא מכוון, ולכן מומלץ ללמידה עצמית.

במהלך הדיון בפרק זה נניח ש־$G=(V,E)$ הוא גרף פשוט ומכוון (בגרף לא מכוון, אין הבדל בין קשירות היטב לקשירות רגילה).

הגדרות: הגרף G יקרא קשיר היטב, אם לכל זוג צמתים $u,v\in V$ קיים מסלול מכוון מ־u ל־v. רכיב קשיר היטב הוא תת־גרף מושרה על קבוצה $U\subseteq V$, קשיר היטב, כאשר U מקסימלית בהכלה ביחס לזה (אין קבוצה $W\subseteq V$ שמכילה־ממש את U ושעבורה תת־הגרף המושרה גם קשיר היטב).

דוגמאות לגרפים קשירים היטב הם הגרף בעל הצומת הבודד, גרף שהוא מעגל מכוון (על כל הצמתים), או גרף שהוא תוצאה של לקיחת גרף קשיר לא מכוון והחלפת כל קשת בו בקשתות מכוונות לשני הכיוונים. כדאי להעיר שבספרות לפעמים קוראים לגרף בעל התכונה הזו "קשיר חזק" (strongly connected).

בגרף קשיר היטב, הרכיב היחידי של קשירות היטב הוא הגרף עצמו. לעומת זאת, בגרף חסר מעגלים, הרכיבים הקשירים היטב הם כל הקבוצות המורכבות מצומת בודד: על מנת להראות שאין רכיב קשיר היטב גדול יותר, שימו לב שאם היה מסלול מכוון מ־u ל־v עבור $u\neq v$ כל שהם, וגם מסלול מכוון מ־v

ל־u (ולפי ההגדרה של קשירות היטב אמורים להיות מסלולים כאלה בתוך הרכיב), אז השרשור שלהם היה נותן מעגל מכוון בגרף.

נחקור עתה את המבנה הכללי של רכיבים קשירים היטב בגרף, ואח"כ נבנה אלגוריתם שמוצא אותו.

למה (זרות הרכיבים): קבוצות הצמתים של שני רכיבים קשירים היטב באותו גרף אינן יכולות להחתך.

הוכחה: נניח שיש לנו קבוצות צמתים U, W של שני רכיבים קשירים היטב בגרף G. לפי ההגדרה שהמדובר בקבוצות מקסימליות, לא יכול להיות שאחת מהן מכילה את השניה. נניח עתה שקיים צומת $v \in U \cap W$, ונראה שבמקרה כזה גם האיחוד $U \cup W$ משרה תת־גרף קשיר היטב, דבר המהווה סתירה למקסימליות.

נניח ש־u, w שני צמתים ב־$U \cup W$. אם שניהם נמצאים ב־U או שניהם נמצאים ב־W, אז לפי ההנחה שהקבוצות האלו מתאימות לרכיבים קשירים היטב אנחנו כבר יודעים שיש מסלול מ־u ל־w. המקרה האחרון הוא שכל אחד מהצמתים נמצא בקבוצה אחרת. נניח (בלי הגבלת כלליות) שמתקיים $u \in U$ ו־$w \in W$. במקרה כזה, לפי ההנחות קיים מסלול P_1 מ־u ל־v שכולו בתוך U, וכן קיים מסלול P_2 מ־v ל־w שכולו בתוך W (הצומת v נמצא גם ב־U וגם ב־W). אם נשרשר את שני המסלולים, נקבל מסלול מ־u ל־w שכולו נמצא ב־$U \cup W$, ובזאת הוכחנו שגם האיחוד משרה גרף קשיר היטב. ∎

מהלמה האחרונה נובע שמשפחת הרכיבים הקשירים־היטב מתאימה לחלוקה של קבוצת הצמתים בגרף: כל צומת $v \in V$ של הגרף נמצא ברכיב כזה (במקרה "הכי גרוע" הרכיב הוא הצומת עצמו), ולכן משפחת הרכיבים היא משפחה של קבוצות זרות המכסות את V.

הגדרה (גרף הרכיבים הקשירים היטב): גרף הרכיבים הקשירים היטב של G הוא הגרף $\hat{G} = (\hat{V}, \hat{E})$, כאשר $\hat{V}$ הוא משפחת הרכיבים הקשירים היטב ב־G, וקשת מ־$U \in \hat{V}$ ל־$W \in \hat{V}$ מוגדרת להיות ב־$\hat{E}$ אם קיימת קשת $uw \in E$ שעבורה $u \in U$ ו־$w \in W$.

הלמות הבאות מרכזיות להבנת גרף הרכיבים הקשירים היטב.

למה (מסלול של רכיבים): אם יש ב־$\hat{G}$ מסלול מ־U ל־W, אז יש ב־G מסלול מכל צומת $u \in U$ לכל צומת $w \in W$, שכולו נמצא באיחוד של קבוצות הצמתים המתאימות למסלול ב־$\hat{G}$.

הוכחה: נניח ש־$U = V_0, V_1, \ldots, V_k = W$ הוא המסלול, וכמו כן נניח שלכל $0 < i \leq k$, הקשת $u_i v_i \in E$ היא קשת מ־V_{i-1} ל־V_i ב־G. הרעיון הכללי הוא לשרשר מסלולים שבתוך כל V_i יעבירו אותנו מהקשת $u_i v_i$ לקשת הבאה $u_{i+1} v_{i+1}$. ההוכחה הפורמלית היא באינדוקציה.

בסיס האינדוקציה הוא $k = 0$, ואז $U = W$ ולפי ההגדרה של קשירות היטב יש לנו את המסלול מ־u ל־w. עבור המעבר, נניח שהוכחנו את הלמה עבור $k - 1$. בפרט הדבר אומר שיש לנו מסלול P באיחוד $\bigcup_{i=0}^{k-1} V_i$ מהצומת $u \in V_0$ אל הצומת $u_k \in V_{k-1}$. אל הקשת הבודדת $u_k v_k$ נתייחס כאל מסלול P', ונשים לב שקיים בתוך V_k מסלול P'' מ־v_k ל־w, כי זהו רכיב קשיר היטב. איחוד שלושת המסלולים $PP'P''$ הוא המסלול המבוקש מ־u ל־w. ∎

מכאן אפשר להסיק איך נראה גרף הרכיבים $\hat{G}$.

למה: גרף הרכיבים הקשירים הוא חסר מעגלים.

הוכחה: אם $V_0, \ldots, V_k$ היה מתאים למעגל מכוון ב־$\hat{G}$ (עם קשת גם מ־V_k ל־V_0), אז יש ב־$\hat{G}$ מסלול מכוון מכל V_i לכל V_j. לכן לפי הלמה הקודמת היה לנו בתוך האיחוד $\bigcup_{i=1}^{k} V_i$ מסלול מכוון של G בין כל שני צמתים שנמצאים באיחוד. זה היה נותן לנו סתירה למקסימליות של הרכיבים, אלא אם כן כל ה־V_i שווים זה לזה. ∎

המסקנה מכאן היא שגרף הרכיבים הקשירים היטב מציג את כל המאפיינים ה"לא־מעגליים" של הגרף. עתה נראה איך חיפוש לעומק מתנהג ביחס לגרף הרכיבים.

למה (תפישת רכיב): אם u הוא הצומת הראשון שמתגלה במהלך חיפוש לעומק מתוך רכיב קשירות היטב $U \in \hat{V}$, אז כל הרכיב יופיע כצאצאים של u, ויתרה מזו צמתי הרכיב יהוו תת־עץ שם (ז"א שהם קבוצה קשירה ביחס לקשתות עץ החיפוש).

הוכחה: כל צמתי U יופיעו כצאצאי u כמסקנה ישירה של הלמה על ראיה קדימה בחיפוש לעומק (כאשר u מתגלה, לכל צמתי U יש מסלולים מ־u בתוך U, שאף אחד מצמתיהם עוד לא התגלה). הסיבה שהצמתים יופיעו כתת־עץ נובעת מכך שהמדובר ברכיב קשיר (ז"א קבוצה מקסימלית ביחס לקשירות היטב): נניח בשלילה ש־$w \in U$ צאצא של של u בעץ החיפוש, אבל שקיים על המסלול ביניהם $v \in U'$ כאשר $U' \neq U$ הוא רכיב קשירות אחר. מכך נובע שיש גם מסלול מ־U ל־U' וגם מסלול מ־U' ל־U, שזה בלתי אפשרי (ראו את הלמה על חוסר המעגלים בגרף הרכיבים הקשירים היטב). ■

כן יכולים להיות צאצאים של u שאינם שייכים לרכיב שלו U, אולם אלו יופיעו "מחוץ" לצמתים של U. מכיוון שהרגע ראינו גם שכל רכיב קשיר היטב מופיע כתת־עץ של עץ החיפוש, אם יש צאצאים של u ששייכים לרכיב אחר U', אז גם רכיב זה יופיע שם בשלמותו.

האלגוריתם שלנו יבצע חיפוש לעומק תוך כדי שהוא מתחזק מחסנית שניה לצד מחסנית החיפוש. מהמחסנית השניה לא נוציא צמתים כאשר סיימנו לחפש את כל הצאצאים שלהם, אלא יותר מאוחר, כאשר יש בראש המחסנית רצף שמהווה רכיב קשירות היטב בשלמותו. אנחנו נשמור גם לכל צומת v בעץ החיפוש את זמן הגילוי הכי מוקדם של צומת על המחסנית החדשה שיש אליו קשת מצאצא של v, שאותו נסמן ב־$L(v)$. במידה ומתקיים $L(v) = k(v)$, ז"א שאין קשתות מהצאצאים לצמתים עם זמן גילוי יותר מוקדם על המחסנית השניה, הדבר יעיד על כך שרכיב הקשירות של v נמצא כולו על המחסנית, והגיע הזמן להוציא אותו.

תוספות לאלגוריתם חיפוש לעומק מכסה־גרף עבור מציאת קבוצות הצמתים של רכיבים קשירים היטב

פלט נוסף: משפחת קבוצות הצמתים של רכיבים קשירים היטב, W.

משתנים נוספים: מחסנית T של צמתים פעילים, פונקציה a של שייכות ל־T, פונקציה L של זמן גילוי מינימלי של צומת פעיל שהוא שכן של צאצא

- **אתחול נוסף:** $W \leftarrow \emptyset$, $T \leftarrow \emptyset$, ולכל $v \in V$ מציבים $a(v) \leftarrow \text{false}$ (אין צורך לאתחל את L כי הוא רלוונטי רק לצמתים שנמצאים ב־T)
- **כאשר** מכניסים צומת v למחסנית S, מכניסים אותו גם ל־T ומציבים $a(v) \leftarrow \text{true}$, $L(v) \leftarrow k(v)$
- **כאשר** השכן ה־$j(v)$ של v נמצא ב־V'
  - יהי w השכן הנ"ל
  - **אם** $a(w) = \text{true}$ אז מציבים $L(v) \leftarrow \min\{L(v), k(w)\}$
- **כאשר** מוציאים צומת v מהמחסנית S

- **אם** $p(v) \neq \mathrm{nil}$ (v הוא לא שורש של אחד העצים), אז עבור האב $u = p(v)$ של v, מציבים $L(u) \leftarrow \min\{L(u), L(v)\}$
- **אם** $L(v) = k(v)$ (צומת עליון ברכיב קשירות היטב), אז
  - תהי U קבוצת כל הצמתים ב־T מראש המחסנית ועד v
  - מוציאים את U מ־T ומעדכנים $a(u) \leftarrow \mathrm{false}$ לכל $u \in U$
  - מציבים $W \leftarrow W \cup \{U\}$ (מוסיפים את הקבוצה U כאיבר במשפחה W)

מבחינת זמן ריצה, גם אחרי התוספות הזמן יישאר $O(|V| + |E|)$. האתחול הנוסף מוסיף רק $O(|V|)$ זמן עבור מילוי המערך a, ובזכות המערך הזה הבדיקה עבור שייכות של צומת לתור T הופכת להיות בזמן $O(1)$. המקום היחידי שנראה שיכולה להיות פעולה עם זמן ארוך היא בעת הוצאת קבוצת צמתים (שיכולה להיות גדולה) מהמחסנית T כאשר מוציאים צומת מ־S. עם זאת, כל צומת יכול להיכנס ל־T רק פעם אחת (הוא נכנס לשם רק כאשר הוא נכנס גם ל־S), ולכן גם סה"כ הפעולות הקשורות בהוצאות מרוכזות של קבוצות צמתים מ־T מסתכמות לזמן של $O(|V|)$.

לפני שנמשיך, כדאי לציין תכונה של המחסנית T.

למה מיידית: הצמתים על המחסנית T מסודרים (מהזנב לראש) לפי סדר גילוי עולה, מכיוון שזהו הסדר שבו הם מוכנסים ל־ T (הוצאת צמתים מהמחסנית לא משנה את הסדר). ∎

הלמה הבאה על ערך המשתנה L מרכזית להוכחת הנכונות של האלגוריתם.

למה (ערך L): כאשר צומת v מוצא ממחסנית החיפוש S, ערך $L(v)$ יהיה הצומת המוקדם ביותר על המחסנית T שיש אליו קשת מצאצא של v, או v עצמו אם אין צומת יותר מוקדם על T עם קשת כזו.

הוכחה: ההוכחה על ערך $L(v)$ היא באינדוקציה לפי סדר ההוצאה של הצמתים מ־S. הבסיס הוא בעת אתחול האלגוריתם, כאשר עוד לא הוצאנו (וגם לא הכנסנו) אף צומת. אם הטענה היתה נכונה עבור כל הצמתים שהוצאו לפני v, אז בפרט היא נכונה עבור כל הבנים של v (חיפוש לעומק לא יוציא את v מהמחסנית S לפני שיצאו כל הבנים שלו בעץ החיפוש). נשים לב עתה ש־$L(v)$ יהיה המינימום של יעדי כל הקשתות היוצאות מ־v לצמתים שנמצאים לפניו ב־T, עם מינימום ערכי $L(w)$ עבור כל w שהוצאנו (ואלו שווים לפי הנחת האינדוקציה לצומת המינימלי שיש אליו קשת מצאצאים שלהם). סה"כ קיבלנו את המינימום על כל הקשתות מצאצאים של v לצמתים קודמים יותר במחסנית T. כדאי לציין כאן שקבוצת הצמתים שנמצאים ב־T לפני v לא תשתנה לפני ש־v יוצא מ־S (צמתים יכולים לצאת מ־T רק כאשר הם יוצאים מ־S, והצמתים היחידים ב־S עם זמן גילוי לפני v הם אבות קדמונים שלו, ולכן הם לא יצאו לפניו). ∎

למה (נכונות ההוצאה מ־T): בעת הוצאת v מ־S, מתקבל $L(v) = k(v)$ אם ורק אם המדובר בצומת הראשון שהתגלה מרכיב קשיר היטב, ובמקרה זה הקבוצה שיוצאת מ־T היא בדיוק הרכיב המכיל את v.

הוכחה: ההוכחה היא באינדוקציה על פעולות הוצאת צמתים מ־S, כאשר הבסיס הוא בעת האתחול. נוכיח את צעד האינדוקציה.

אם v הצומת הראשון מרכיב קשיר היטב, אז לפי הלמה על תפישת רכיב, כל הרכיב הקשיר של v נמצא בצאצאים שלו בעץ החיפוש. נסמן את הרכיב הזה ב־U. עתה כל $w \in U$ פרט ל־v חייב לקיים $L(w) < k(w)$: לפי הלמה על זרות הרכיבים ולפי הנחת האינדוקציה (שיצאו עד עכשיו רק רכיבים קשירים היטב בשלמותם) עוד לא יצאו מ־T צמתים של U. על כן חייה להיות צאצא ל־w עם קשת

לצומת יותר מוקדם ב־T, אחרת לא יכול להיות ממנו מסלול ל־v שמוכל ב־U. מכאן שצמתי הרכיב U לא יצאו מ־T לפני ש־v יוצא מ־S.

לעומת זאת, אם היה ל־v בעץ החיפוש צאצא ששייך לרכיב קשירות אחר U', אז כל צמתי U' יהיו צאצאים של v כי הם מהווים תת־עץ של עץ החיפוש (החלק השני של הלמה על תפישת רכיבים קשירים). בפרט זה כולל את הצומת v' שהוא הצומת הראשון שהתגלה מ־U', ולכן לפי הנחת האינדוקציה הרכיב U' יצא בשלמותו כאשר v' יצא מ־S, כי זה קרה לפני ש־v יצא מ־S.

סה"כ קיבלנו שכאשר v יוצא מ־S, יש ב־T מתחתיו בדיוק את צמתי רכיב הקשירות U. על מנת לסיים נותר להוכיח שאכן $L(v) = v$: אם זה לא היה המצב, הרי שהיתה קשת מצאצא w של v, לצומת מוקדם יותר ב־T. צומת כזה חייב להיות ברכיב קשירות U' שונה מ־U. אם U' לא מכיל אב קדמון של v, אז כל צמתי U' כבר הוצאו מ־S לפני ש־v הוכנס אליו (המחסנית S של חיפוש לעומק תמיד מכילה מסלול), ולכן לפי הנחת האינדוקציה רכיב זה כבר הוצא מ־T לפני הכנסת v וקיבלנו סתירה. אם U' כן מכילה אב קדמון של v, אז קיבלנו סתירה למקימליות הרכיב: יהיה אפשר להוסיף ל־U את האיחוד של המסלול שלו בעץ החיפוש ל־w, את הקשת מ־w לרכיב U', ומשם את המסלול לאב הקדמון של v ול־v עצמו. ■

מהלמה האחרונה נובע שכאשר סיימנו את החיפוש (וכיסינו את כל צמתי הגרף), המשפחה W תהיה משפחת כל רכיבי הקשירות של G.

שימוש בחיפוש לעומק עבור בניית גרף רכיבים אי־פריקים

הערה: חלק מהטענות כאן מוצגות ללא הוכחות מלאות – הוכחות מלאות היו לוקחות מקום רב מדי (והמדובר יותר בקומבינטוריקה מתמטית מאשר במדעי המחשב). מומלץ כהכנה לכאן לקרוא את הפרק הקודם על האלגוריתם של Tarjan לרכיבים קשירים היטב, כי הוא מכיל הוכחות מלאות וחלק מהרעיונות שם דומים לאלו כאן.

במהלך דיון זה נניח ש־$G = (V, E)$ הוא גרף פשוט, לא מכוון וקשיר עם לפחות שני צמתים. אנחנו נגדיר עבורו מושג של "אי־פריקות" (בעצם 2־קשירות בצמתים), ונראה אלגוריתם למציאת "פירוק" של גרף קשיר לתתי־גרפים כאלו. ראשית, קצת קומבינטוריקה.

הגדרות: צומת $u \in V$ יקרא צומת מפריד אם קיימים $v, w \in V$ שונים מ־u כך שכל מסלול ביניהם חייב לעבור דרך u. במילים אחרות, u מפריד אם ורק אם $G[V \setminus \{u\}]$, תת־הגרף של G המושרה על $V \setminus \{u\}$, אינו קשיר (בניגוד ל־G עצמו).

גרף יקרא אי־פריק אם אין בו צמתים מפרידים, ורכיב אי־פריק (בגרף קשיר כל שהוא) הוא תת־גרף מושרה אי־פריק (וקשיר) מקסימלי ביחס להכלה של קבוצת הצמתים.

למת מבנה שתועיל בהמשך: כל קשת בגרף G מוכלת ברכיב אי־פריק כל שהוא (ובפרט אין רכיבים אי־פריקים של צומת בודד). כמו כן, שני רכיבים אי־פריקים שונים של G לא יכולים להיחתך ביותר מצומת בודד (ולכן כל קשת מוכלת ברכיב אי־פריק יחידי), ואם שני רכיבים אי־פריקים נחתכים בצומת בודד אז הוא צומת הפרדה.

הוכחה: שני צמתים עם קשת ביניהם מהווים בפרט גרף אי־פריק, ולכן אם vw היא קשת בגרף אז $\{v, w\}$ מוכלת ברכיב אי־פריק (או מהווה אחד בעצמה).

אם יש שני רכיבים אי־פריקים שונים W, W' הנחתכים ביותר מצומת אחד, אז ניתן להראות ש־$W \cup W'$ גם משרה תת־גרף אי־פריק, בסתירה להגדרה של המקסימליות של הרכיבים, בצורה הבאה: תת־הגרף המושרה על $W \cup W'$ הוא קשיר, כי בפרט W ו־W' קשירים ויש להם צמתים משותפים. עתה

מראים עבור כל $v \in W \cup W'$ שתת־הגרף המושרה $G[W \cup W' \setminus \{v\}]$ גם קשיר. מכיוון ש־W אי־פריק, תת־הגרף המושרה $G[W \setminus \{v\}]$ קשיר (בין אם $v \in W$ ובין אם לאו), ואותו דבר נכון עבור $G[W' \setminus \{v\}]$. לפי הנחת השלילה קיים צומת $u \in W \cap W' \setminus \{v\}$. עתה עבור שני צמתים ב־$W \cup W' \setminus \{v\} = (W \setminus \{v\}) \cup (W' \setminus \{v\})$ נבנה מסלול מקשר בצורה הבאה: אם שניהם ב־$W \setminus \{v\}$ או שניהם ב־$W' \setminus \{v\}$ אז המסלול נובע מהקשירות של תת־הגרף המושרה המתאים. אם צומת אחד מהם ב־$W \setminus \{v\}$ והצומת השני ב־$W' \setminus \{v\}$, ניקח מסלול ב־$W \setminus \{v\}$ מהצומת הראשון ל־u, ונחבר אותו למסלול ב־$W' \setminus \{v\}$ מ־u לצומת השני.

אם שני הרכיבים נחתכים בצומת בודד u, אז ניקח צמתים $w \in W \setminus \{u\}$ ו־$w' \in W' \setminus \{u\}$. אם היה ביניהם מסלול P שאינו עובר דרך u, אז $W \cup W' \cup P$ היה משרה תת־גרף אי־פריק, בסתירה למקסימליות הרכיבים. להוכחת אי הפריקות מניחים (בלי הגבלת כלליות) שרק צמתי הקצה של P נמצאים ברכיבים W ו־W' בהתאמה. מכאן ההמשך דומה לסעיף הקודם, רק שצריך לפצל את הניתוח של $G[W \cup W' \cup P \setminus \{v\}]$ למקרה שבו $v \in P$ (מכיוון שאז $v \neq u$, מסלולים בין צמתים יעברו דרך $W \cup W' \setminus \{v\}$, שמשרה תת־גרף קשיר), ולמקרה שבו $v \notin P$ (המקרה הבעייתי זה כאשר $v = u$; בכל מקרה גם $W \setminus \{v\}$ וגם $W' \setminus \{v\}$ ישרו תתי־גרף קשירים, ועבור צומת מ־W וצומת מ־W' אפשר להשלים מסלולים מקשרים דרך המסלול P). ■

נגדיר את גרף הרכיבים האי־פריקים של G כגרף הדו־צדדי הבא על קבוצת הצמתים $U \cup W$: נגדיר את U להיות קבוצת צמתי ההפרדה של G, את W להיות קבוצת הרכיבים האי־פריקים של G, ולכל $u \in U$ ו־$w \in W$ (שימו לב ש־w הוא תת־קבוצה של V) נגיד ש־uw קשת אם $u \in w$. בפרט אם G כולו אי־פריק, אז הגרף הזה מכיל צומת יחיד ב־W ולא מכיל צמתים ב־U.

טענה: הגרף הנ"ל הוא עץ.

הוכחה: על מנת להראות קשירות, לכל זוג צמתים $u, u' \in U$ נוכל לקחת את המסלול ביניהם ב־G (כי אלו בפרט צמתים ב־G, שהוא גרף קשיר). במסלול זה נשאיר את הצמתים המפרידים, ונחליף כל קשת ברכיב האי־פריק (היחידי) המכיל אותה לפי למת המבנה; למעשה כל קטע מסלול בין שני צמתים מפרידים עוקבים יתאים לרכיב אי־פריק יחיד, כי לפי למת המבנה הרכיב יכול "להתחלף" רק כאשר עוברים דרך צומת מפריד. זה נותן לנו קיום מסלול בין כל זוג צמתים ב־U. עבור זוגות צמתים אחרים אפשר לפעול באופן דומה, למשל עבור זוג $w, w' \in W$ ניקח מסלול ב־G שמתחיל בקשת פנימית לרכיב w ומסתיים בקשת פנימית לרכיב w', ונתרגם אותו למסלול בגרף הרכיבים.

על מנת להראות שאין מעגלים, נניח בשלילה ש־C הוא מעגל פשוט בגרף הרכיבים. מעגל כזה צריך להכיל לפחות שני צמתים מ־W (כי הגרף דו־צדדי), ז"א שני רכיבים אי־פריקים שונים. נסתכל אז ב־G על קבוצת הצמתים $(C \cap U) \cup \bigcup_{w \in C \cap W} w$. במילים, זהו איחוד קבוצת הצמתים המפרידים המופיעים ב־C עם תוכן רכיבי הקשירות המופיעים ב־C. אפשר להראות שקבוצה זו משרה תת־גרף אי־פריק, בסתירה למקסימליות, בצורה מאוד דומה להוכחה של למת המבנה הקודמת. ■

עכשיו כשאנחנו יודעים את הקומבינטוריקה של עצי הרכיבים האי־פריקים, נרצה לפתח אלגוריתם שימצא לנו אותם. ראשית נתמקד במציאת צמתים מפרידים, עבורם עצי חיפוש לעומק יהיו מועילים ביותר. נתחיל מצומת ההתחלה של החיפוש.

הגדרה: עבור צומת v בעץ חיפוש (שמתקבל מחיפוש לעומק או מאלגוריתם חיפוש אחר), נסמן ב־T_v את תת־העץ המורכב מכל הצאצאים של v (כולל v עצמו).

למה (אפיון שורש מפריד): השורש של עץ חיפוש לעומק הוא צומת מפריד אם ורק אם יש לו לפחות שני בנים.

הוכחה: בכיוון אחד, אם אין לשורש s שני בנים, אז לכל שני צמתים אחרים בגרף ניתן למצוא מסלול ביניהם שאינו עובר דרך s תוך שימוש בקשתות העץ בלבד.

מצד שני, נניח שיש ל־s שני בנים u ו־v. מכיוון שאין בעץ חיפוש לעומק של גרף לא־מכוון קשתות חוצות. לא יהיו קשתות ב־G בין תתי־העצים המתאימים T_u ו־T_v. כמו כן, אם יש עוד בנים ל־s, לא יהיו גם קשתות מתתי־העצים שלהם לאף תת־עץ אחר המתאים לבן של s. על כן, אם נסיר את s, לא יהיה מסלול בגרף הנשאר בין u ל־v. ■

היינו יכולים עכשיו למצוא את הצמתים המפרידים ע"י הרצת חיפוש לעומק מכל צומת בגרף, אבל אנחנו נרצה שיטה יעילה יותר, שבה עושים את החיפוש פעם אחת בלבד. ראשית נראה שלא צריך לדאוג לצמתים עלים בעץ ה־חיפוש לעומק.

טענה מיידית: עלה בעץ חיפוש לעומק הוא לעולם אינו צומת מפריד, מכיוון שאם מסירים אותו אז שאר הצמתים נשארים מקושרים דרך קשתות העץ עצמו. ■

בשביל שאר הצמתים נשתמש בעוד הגדרה, ולמה שמכלילה את הטענה המיידית למעלה.

הגדרה: קשת בגרף מצאצא של u לאב קדמון של u (שניהם לא u עצמו) תיקרא קשת עוקפת (זוהי בפרט קשת לאחור).

למה (אפיון צמתים מפרידים): צומת u בעץ חיפוש לעומק שאינו שורש הוא מפריד אם ורק אם יש לו בן v שעבורו אין קשת מ־T_v שעוקפת את u.

הוכחה: בכיוון אחד, אם עבור בן v אין קשת מ־T_v שעוקפת את u אז זה אומר שאין מ־T_v קשתות כלל לצמתים אחרים בגרף פרט ל־u (עבור צמתים שאינם אבות קדמונים של u הדבר נובע מכך שאין קשתות חוצות). על כן בפרט אין בגרף מסלול מ־v לשורש s שאינו עובר דרך u.

בכיוון שני, אם לבן v יש קשת עוקפת $v'w$ מ־T_v לאב קדמון של u, אז נראה איך אפשר למצוא מסלול מכל צומת של T_v לשורש s של עץ החיפוש. מזה ינבע שאם לכל הבנים יש מסלולים עוקפים כאלו, אז הגרף המושרה על $V \setminus \{u\}$ הוא קשיר (ולכן u אינו צומת מפריד), כי מצמתים שאינם צאצאים של u גם יש מסלול ל־s שאינו עובר דרך u, ע"י שימוש בקשתות העץ בלבד.

בהינתן הבן v והקשת העוקפת $v'w$, לכל צומת $r \in T_v$ נמצא רק באמצעות קשתות העץ T_v מסלול ל־v', משם ניקח את הקשת $v'w$, ומ־w ניקח את המסלול ל־s על קשתות עץ החיפוש. ■

לפני שנמשיך, כדאי לציין גם את המסקנה הבאה (ההוכחה שלה מתוך מה שהוכח למעלה מושארת לכם כתרגיל), אשר תגדיר לנו בדיוק באיזו צורה ההורדה של צומת מפריד מפרקת את הגרף.

מסקנה (רכיבים לאחר הוצאת צומת מפריד): אם u הוא צומת מפריד, יהיו $u_1, \ldots, u_k$ כל הבנים שלו ללא קשתות עוקפות, ואם $u = s$ אז ניקח את $u_1, \ldots, u_k$ להיות כל הבנים. רכיבי הקשירות של תת־הגרף המושרה על $V \setminus \{u\}$ יהיו קבוצות הצמתים של $T_{u_1}, \ldots, T_{u_k}$, בנוסף לקבוצת הצמתים הנותרים ב־$V \setminus \{u\}$ לאחר חיסור כל אלו (אם $u \neq s$). ■

על מנת להפוך את הידע שלנו לאלגוריתם, צריך עוד למצוא שיטה לשלב את החיפוש לעומק בתחזוק של מידע על מסלולים עוקפים. אנחנו נשתמש בכך שהאלגוריתם כבר כותב לכל צומת את זמן הגילוי שלו (הערך $k(v)$), ונתחזק לכל צומת v ערך $L(v)$ שיכיל לבסוף את זמן הגילוי המוקדם ביותר של צומת שיש אליו קשת מ־T_v (במקרה הקצה של השורש נגדיר $L(s) = k(s) = 1$). בצורת נוסחה, $L(v) = \min\{k(w) : u \in T_v,\, uw \in E\}$. לפני שנראה איך מתחזקים את $L(v)$, נראה מדוע זה עוזר לנו.

טענה: צומת u שאינו שורש יהיה מפריד אם ורק אם קיים לו בן v שעבורו $L(v) = k(u)$.

הוכחה: בגלל שאין קשתות חוצות בעץ חיפוש לעומק, תמיד $L(v)$ יתאים לאב קדמון של v. זה יהיה האב u אם אין קשתות שעוקפות אותו מתוך T_v, ואחרת זה יתאים לצומת שהתגלה קודם לכן – שימו לב שתמיד לאבות הקדמונים של u יהיה זמן גילוי נמוך מ־u עצמו. הטענה נובעת מהפעלת הניתוח הזה על כל הבנים של u, יחד עם הלמה על אפיון צמתים מפרידים בעץ חיפוש לעומק. ■

לאור זאת, מציאת כל הצמתים המפרידים תהיה באמצעות הרצת חיפוש לעומק עם תחזוק של $L(v)$, שלאחריו בודקים האם לשורש s יש יותר מבן אחד (בשביל לדעת אם לסמן אותו כצומת מפריד), ולשאר הצמתים שאינם עלים בודקים את ערכי $L(v)$ של בניהם (ובהתאם לכך מסמנים אותם).

התוספות הבאות לאלגוריתם חיפוש לעומק מתחזקות את $L(v)$.

תוספות לחיפוש לעומק (עם מחסנית) בשביל לחשב את ערכי $L(v)$

משתנים נוספים: פונקציה L של זמן גילוי מינימלי של צומת שיש אליו קשת מצאצא

- **כאשר** מציבים את ערך $k(v)$ כאשר v מוכנס למחסנית, מציבים גם $L(v) \leftarrow k(v)$
- **כאשר** השכן ה־$j(v)$ של v נמצא ב־V'
  - יהי w השכן הנ"ל
  - מציבים $L(v) \leftarrow \min\{L(v), k(w)\}$
- **כאשר** מוציאים צומת $s \neq v$ מהמחסנית
  - עבור האב $u = p(v)$ של v, מציבים $L(u) \leftarrow \min\{L(u), L(v)\}$

עבור טענת הנכונות ניתן רק סקיצה של ההוכחה.

טענה (נכונות החישוב): כאשר v יוצא מהמחסנית, $L(v)$ יכיל את הערך הנכון עבורו.

סקיצת הוכחה: ההוכחה באינדוקציה על גובה תת־העץ T_v, כאשר בסיס האינדוקציה הוא כאשר v הוא עלה בעץ ה־חיפוש לעומק. עבור v שהוא עלה האלגוריתם יתן $L(v) = \min\{k(w) : vw \in E\}$, שזהו הערך הנכון עבורו. בהינתן v שהוא אינו עלה, מניחים באינדוקציה שבכל פעם שמוציאים ממחסנית החיפוש בן w של v, בן זה יקבל את הערך הנכון של $L(w)$. התוספת לאלגוריתם בעת היציאה מ־w גורמת לכך שבעת ההוצאה של v, הערך של $L(v)$ יהיה המינימום מבין כל ערכי ה־L של בניו וכל ערכי זמן הגילוי של הצמתים שהוא עצמו מחובר אליהם, וזה יהיה הערך הנכון עבור $L(v)$. ■

השאלה האחרונה שיכולה להישאל היא האם ניתן למצוא ביעילות גם את הרכיבים האי־פריקים ולא רק את הצמתים המפרידים. התשובה היא חיובית, ומסתמכת על המסקנה מלמעלה באשר לרכיבי הגרף הנותרים כשמוציאים ממנו צומת מפריד. האלגוריתם למצוא צמתים מפרידים ורכיבים אי־פריקים תוך כדי ביצוע חיפוש לעומק נתון ע"י התוספות הבאות לאלגוריתם (בנוסף לאלו המתחזקות את ערכי $L(v)$).

תוספות לחיפוש לעומק, בנוסף להוספת חישוב $L(v)$, למציאת צמתים מפרידים ורכיבים אי־פריקים

פלט נוסף: רשימת צמתים מפרידים U, רשימת קבוצות רכיבים אי־פריקים W

משתנים נוספים: מחסנית נוספת T שמחזיקה תתי־קבוצה של V

- **אתחול נוסף:** $U \leftarrow \emptyset$, $W \leftarrow \emptyset$, ומאתחלים את T להחזיק קבוצה יחידה $\{s\}$
- **כאשר** מכניסים צומת חדש w למחסנית החיפוש S
  - דוחפים את הקבוצה $\{w\}$ למחסנית הקבוצות T
- **כאשר** מוציאים צומת $w \neq s$ מהמחסנית S (לאחר מיצוי כל שכניו)
  - **אם** w אינו הבן הראשון של s (אפשר לבדוק לפי $k(w)$) וגם $L(w) = k(p(w))$ (אין מסלול עוקף מתת־העץ T_w, כולל את המקרה $p(w) = s$)
    - מוציאים את הקבוצה העליונה R מהמחסנית T ומוסיפים את $R \cup \{p(w)\}$ ל־W
    - מוסיפים את $p(w)$ ל־U (אם לא הוסף לשם קודם)
  - **אם** w הבן הראשון של s או $L(w) < k(p(w))$ (יש מסלול עוקף)
    - מוציאים את הקבוצה העליונה R מהמחסנית T
    - מחליפים את הקבוצה העליונה הנוכחית R' במחסנית T באיחוד $R \cup R'$
- **כאשר** מוציאים את s מהמחסנית S
  - מוסיפים את הקבוצה העליונה (והיחידה) במחסנית T ל־W

לא ניתן כאן הוכחה מדוע זה עובד. קצת אינטואיציה: הבניה של U היא בדיוק לפי הדיון למעלה בצמתים מפרידים. באשר ל־W, קבוצת צמתים תהיה מוכלת ברכיב רק אם בין כל שניים מהם יש מסלול שאינו עובר (פרט אולי לצמתי הקצה עצמם) דרך צומת מפריד. בפרט, צומת מפריד יימצא בכל הרכיבים המכילים שכן לא־מפריד שלו לפי עץ החיפוש (כל אלו למות הדורשות הוכחה – אתם מוזמנים להוכיח אותן כתרגיל), וצמד אב־בן של צמתים מפרידים על עץ החיפוש יהוו בעצמם רכיב.

אתם גם מוזמנים להראות כתרגיל שאפשר לממש את התחזוק של T (והקבוצות בתוכה) בצורה שתוסיף זמן חישוב של $O(1)$ בלבד לכל צומת של הגרף.

עצים פורשים מינימלים

הגדרות והאלגוריתם הגנרי

בבעיה של עץ פורש מינימלי, נתון לנו גרף לא מכוון, פשוט וקשיר $G = (V, E)$, וכן נתונה פונקצית משקל $w : E \to \mathbb{R}^+$ (ליתר דיוק, אברי הטווח של w צריכים להיות אוביקטים מתמטים שניתן לבצע עליהם

פעולות חיבור והשוואה במהירות – למשל מספרים טבעיים חסומים ע"י פולינום נתון ב־$|V|$). המטרה היא למצוא עץ פורש $T = (V, E')$ של הגרף שעבורו $\sum_{e\in E'} w(e)$ הוא המינימלי ביותר האפשרי. קבוצת כל העצים הפורשים את הגרף היא סופית (חסומה ע"י קבוצת כל תתי־הקבוצות של E) ולא ריקה (הנחנו ש־G קשיר), ולכן קיים עץ פורש המשיג את המינימום.

מדוע נרצה למצוא כזה עץ? אפשר לחשוב על זה כמודל לבעיה שבה מנסים לחבר רשת תקשורת (שצריכה להיות קשירה) במינימום אפשרי של עלות. וכמו רב האלגוריתמים כאן, גם אלגוריתם לעץ פורש מינימלי יוכל לשרת אתכם בהמשך כרכיב באלגוריתמים יותר מורכבים.

עתה נראה אלגוריתם גנרי לבעית העץ הפורש המינימלי, שבו אין הבטחה טובה לזמן ריצה. אבל כפי שנראה בהמשך, לנכונות של האלגוריתם הגנרי כאן יש עוצמה חזקה במיוחד – לא רק שהדבר יסייע לנו בהוכחת הנכונות של אלגוריתמים אחרים, אלא נוכל גם להוכיח טענת אפיון קומבינטורית לעצים פורשים מינימלים מתוך הנכונות של האלגוריתם הגנרי.

האלגוריתם הגנרי מתחיל ממצב שבו כל הקשתות צבועות ב"לבן", ובכל שלב האלגוריתם יכול לצבוע קשת ב"כחול", ז"א קשת שבטוח תהיה בעץ שהאלגוריתם יפלוט לבסוף, או ב"אדום", ז"א קשת שבטוח לא תהיה בעץ הפלט (אם שתי הפעולות אפשריות, אז האלגוריתם יכול לבחור אחת מביניהם באופן שרירותי).

רגע לפני הצגת האלגוריתם, נזכיר שחתך בגרף G מוגדר ע"י זוג של קבוצות צמתים זרות לא ריקות (U, W) שאיחודן הוא V (ז"א $W = V \setminus U$), וקשתות החתך מוגדרות להיות קבוצת כל הקשתות ב־E שיש להן צומת מ־U וצומת מ־W. אפשר לסמן קבוצה זו כך: $E \cap (U \times W)$, ז"א החיתוך של E עם קבוצת הזוגות של צומת מ־U וצומת מ־W. כדאי לשים לב כבר עכשיו שבהינתן חתך (U, W) ומעגל C, מספר הקשתות המשותפות למעגל ולחתך חייב להיות זוגי, ובפרט אם יש קשת משותפת כזו אז יש לפחות שתיים מהן. אנחנו נשתמש באבחנה הזו הרבה בהמשך.

אלגוריתם גנרי למציאת עץ פורש מינימלי

קלט: גרף קשיר ולא מכוון $G = (V, E)$, פונקצית משקל $w : E \to \mathbb{R}^+$

פלט: עץ פורש T עם משקל מינימלי

- **אתחול:** מציבים $c(e) \leftarrow \text{white}$ לכל $e \in E$
- **כל עוד** אפשר לבצע לפחות אחת מהפעולות הבאות
  - **אם** יש ב־G מעגל C שאף קשת בו לא מקיימת $c(e) = \text{red}$
    - תהי $e \in C$ קשת לבנה עבורה $w(e)$ מקסימלי מבין קשתות המעגל הלבנות
    - מציבים $c(e) \leftarrow \text{red}$ **(הכלל האדום)**
  - **אם** יש ב־G חתך (U, W) כך שאף קשת בו לא מקיימת $c(e) = \text{blue}$
    - תהי $e \in E \cap (U \times W)$ קשת לבנה בחתך עבורה $w(e)$ מינימלי מבין הנ"ל
    - מציבים $c(e) \leftarrow \text{blue}$ **(הכלל הכחול)**

• מגדירים $E_b = \{e \in E : c(e) = \text{blue}\}$ ומגדירים עבור הפלט $T = (V, E_b)$

זמן הריצה של האלגוריתם יכול להיות גדול מאוד ללא פרטים נוספים על איך מוצאים מקרים שמקיימים את הכלל האדום או הכלל הכחול (האלגוריתמים הספציפיים שנראה מוצאים אותם ביעילות). בהמשך נוכיח שלכל סדרה של הפעלות של הכללים נישאר לבסוף עם פלט שהוא עץ פורש מינימלי. על מנת לתת יותר אינטואיציה על איך הדבר נעשה, נוכיח קודם כל רק "שפיות" בסיסית, ז"א שהאלגוריתם תמיד יצליח להפעיל את הכללים עד שכל הקשתות צבועות, ושהוא תמיד יחזיר עץ פורש. את זאת נקבל כתוצאה מסדרת הטענות הבאה.

טענה (חוסר חסימה): אם (U, W) חתך בגרף שאין בו קשתות כחולות, ורק קשת בודדת בו היא לבנה (בגלל שהשאר אדומות או בגלל שהיא הקשת היחידה בחתך), אז היא לא תיצבע באדום.

הוכחה: נסמן את הקשת שלנו ב־uw כאשר $u \in U$ ו־$w \in W$. כל מעגל המכיל את uw חייב להכיל לפחות קשת אחת נוספת $u'w'$ מהחתך (U, W). אולם לפי ההנחות שלנו הקשת הזו (אם היא בכלל קיימת ב־E) חייבת להיות אדומה, ואז לא ניתן להכיל על המעגל את הכלל האדום. ■

מסקנה: במהלך הרצת האלגוריתם לעולם לא יהיה חתך שכל קשתותיו אדומות, מכיוון שלעולם לא נוכל לצבוע באדום את הקשת האחרונה בו.

נסמן עתה ב־E_b את קבוצת הקשתות הכחולות.

טענה (קשירות): כל עוד E_b אינה מתאימה לתת־גרף קשיר פורש של G, ניתן לבצע על קשת כל שהיא את הכלל הכחול. בפרט האלגוריתם יכול לעצור רק אם E_b קשיר.

הוכחה: נגדיר את U להיות רכיב קשירות של (V, E_b), ובהתאמה נגדיר $W = V \setminus U$. עתה (U, W) הוא חתך אשר אינו מכיל קשתות כחולות (והוא מכיל קשתות כי הגרף קשיר). לפי המסקנה מטענת חוסר החסימה, תהיה בו קשת לבנה אחת לפחות, ואז נוכל להפעיל עליו את הכלל הכחול ולצבוע קשת מתוכו. בפרט זה אומר שהאלגוריתם לא יעצור בשלב הזה (האלגוריתם עוצר רק כשאין אף פעולה אפשרית). ■

טענה (השלמת צביעה): אם E_b מתאימה לתת־גרף קשיר פורש, אז ניתן לצבוע את שאר הקשתות הלבנות באדום.

הוכחה: עבור קשת לבנה uw, נחבר אותה למסלול של קשתות כחולות מ־u ל־w (יש כזה לפי הנחת הקשירות), ונקבל מעגל ללא קשתות אדומות שבו uw היא הקשת הלבנה היחידה. על כן נוכל לצבוע אותה לפי הכלל האדום. ■

מטענה הקשירות וטענת השלמת הצביעה, נובע שהאלגוריתם תמיד יעצור ותמיד יתן תת־גרף קשיר פורש. על מנת להראות שזה יהיה עץ, נותר עוד להוכיח טענה קלה אחרונה על קבוצת הקשתות E_b.

טענה (אי סגירת מעגלים): הכלל הכחול לעולם לא יסגור מעגל של קשתות כחולות, ולכן E_b תישאר חסרת מעגלים (ובסוף האלגוריתם תתאים לעץ פורש).

הוכחה: אם uw היא קשת בחתך (U, W) אשר סוגרת מעגל C של קשתות כחולות, אז חייבת להיות ב־C קשת נוספת מתוך החתך (U, W). אבל אז הדבר אומר שאיננו יכולים להפעיל את הכלל הכחול עבור החתך הנ"ל. ■

עתה ניגש לעיקר, וזו ההוכחה שהאלגוריתם הגנרי אכן נותן לנו עץ פורש מינימלי. ראשית ניתן מספר תנאים הכרחיים שמתקיימים לכל עץ פורש מינימלי. נקבע לנו עץ פורש $T = (V, E')$ של הגרף. עבור

קשת $e \in E'$, היער $(V, E' \setminus \{e\})$ הוא בעל שני רכיבי קשירות בדיוק. נסמן אותם ב־U_e, W_e (לא משנה לנו הסדר), ונגדיר לפיהם את החתך בגרף (U_e, W_e).

טענה (מינימליות בחתך): אם $T = (V, E')$ עץ פורש מינימלי, ו־$e \in E'$ קשת מתוכו, אז e היא בעלת משקל מינימלי מבין קשתות החתך (U_e, W_e). בנוסף, אם $f \neq e$ היא קשת אחרת בעלת משקל מינימלי מהחתך (U_e, W_e), אז הגרף $T' = (V, E' \cup \{f\} \setminus \{e\})$ הוא גם עץ פורש מינימלי.

הוכחה: קודם נניח בשלילה ש־e אינה בעלת משקל מינימלי, ונסמן ב־f את הקשת בעלת המשקל המינימלי (שימו לב שמההגדרה של החתך לפי רכיבי קשירות נובע בפרט ש־$f \notin E'$). אם מסתכלים על $T' = (V, E' \cup \{f\} \setminus \{e\})$, אז הוא חסר מעגלים (אין מעגל המכיל את f כי היא קשת יחידה בחתך, ולא היו מעגלים אחרים בתוך $E' \setminus \{e\}$), ולכן הוא עץ (כי גם לו יש $|V| - 1$ קשתות). בנוסף, המשקל של T' קטן מהמשקל של T בהפרש $w(e) - w(f)$, סתירה למינימליות T.

עבור החלק השני של הטענה, אם e בעלת משקל מינימלי בחתך ו־f קשת שניה בעלת אותו משקל מינימלי, אז כמו בטיעון הקודם $T' = (V, E' \cup \{f\} \setminus \{e\})$ הוא עץ פורש, וחישוב המשקל שלו יתן משקל זהה לזה של העץ המינימלי T. ■

לפני שנמשיך, מסקנה מעניינת.

מסקנה: כל עץ מינימלי של G יכול להיות מוצג כפלט של הרצה מתאימה של האלגוריתם הגנרי.

הוכחה: לשם כך ראשית נצבע כל קשת $e \in E'$ בכחול, לפי הפעלת הכלל הכחול עם החתך (U_e, W_e), תוך שימוש בטענת המינימליות לפי חתך. אחרי זה נצבע את שאר הקשתות באדום לפי טענת השלמת הצביעה שהראינו קודם. ■

כמובן, המטרה שלנו היא להראות שרק עצים פורשים מינימלים ניתנים להצגה כפלט של האלגוריתם הגנרי, וכאן הראינו את הכיוון ההפוך. נגדיר עוד מבנה שקשור בעצים וקשתות: עבור העץ הפורש $T = (V, E')$ וקשת $e \notin E'$, יהי C_e המעגל הפשוט המתקבל מהחיבור של e עם המסלול הפשוט היחידי ב־E' בין שני הצמתים של e.

טענה (מקסימליות במעגל): אם $T = (V, E')$ עץ פורש מינימלי, ו־$e \notin E'$ קשת שאינה בעץ, אז e היא בעלת משקל מקסימלי מבין הקשתות של C_e. בנוסף, אם $f \neq e$ היא קשת אחרת בעלת משקל מקסימלי מהמעגל C_e, אז הגרף $T' = (V, E' \cup \{e\} \setminus \{f\})$ הוא גם עץ פורש מינימלי.

הוכחה: נניח בשלילה ש־e אינה בעלת משקל מקסימלי, ונסמן ב־f את הקשת בעלת המשקל המקסימלי (שימו לב שמההגדרה של C_e נובע בפרט ש־$f \in E'$). אם מסתכלים על $T' = (V, E' \cup \{e\} \setminus \{f\})$ אז הוא קשיר (מושאר כתרגיל) ולכן הוא עץ (אותו מספר קשתות). בנוסף לכך המשקל שלו קטן מזה של T בהפרש $w(f) - w(e)$, וזו סתירה.

עבור החלק השני, אם e בעלת משקל מקסימלי במעגל ו־f קשת שניה בעלת אותו משקל מקסימלי, אז כמו בטיעון הקודם $T' = (V, E' \cup \{e\} \setminus \{f\})$ הוא עץ פורש, וחישוב המשקל שלו יתן משקל זהה לזה של העץ המינימלי T. ■

עתה אפשר להוכיח את המשפט המרכזי.

משפט הנכונות של האלגוריתם הגנרי: בסוף כל הרצה אפשרית של האלגוריתם הגנרי נקבל עץ פורש מינימלי של הגרף.

הוכחה: מראים, באינדוקציה על מספר הצעדים שנעשו, את השמורה הבאה: בכל שלב של האלגוריתם קיים עץ פורש מינימלי $T = (V, E')$, כך שקבוצת הקשתות הכחולות E_b מוכלת ב־E', בעוד שקבוצת הקשתות האדומות E_r זרה ל־E'. מכאן שבסוף האלגוריתם הקבוצה E_b תהיה שווה בדיוק לקבוצת

הקשתות של עץ פורש מינימלי (כי כבר הוכחנו שהסוף מגיע רק כש־E_b היא קבוצת הקשתות של עץ פורש כל שהוא).

הבסיס הוא תחילת האלגוריתם שבו $E_b = E_r = \emptyset$. אז כל עץ פורש מינימלי $T = (V, E')$ יתאים לתנאי של השמורה. עבור צעד המעבר, נניח שלפני הצעד ה־i היה קיים עץ T כזה, ונראה שקיים עץ פורש מינימלי T' (שיכול להיות שונה מ־T) שיקיים את השמורה לאחר הצעד ה־i. קיימים שני מקרים לפי הכלל שהשתמשנו בו בצעד זה.

שימוש בכלל הכחול: נניח שצבענו את הקשת e בכחול לפי החתך (U, W). אם $e \in E'$ אז מגדירים $T' = T$ ואין בעיה. אם $e \notin E'$, אז נבדוק את המעגל C_e. למעגל הזה יש את הקשת e המשותפת עם החתך, ולכן חייבת להיות למעגל ולחתך (U, W) לפחות עוד קשת משותפת אחת f. מתקיים $w(e) \leq w(f)$, כי e היא בעלת משקל מינימלי בחתך לפי הגדרת הכלל הכחול. בנוסף, לפי טענת המקסימליות במעגל C_e מתקיים $w(f) \leq w(e)$, ז"א ששתי הקשתות מקבלות את המשקל המקסימלי במעגל C_e. על כן לפי אותה טענה גם העץ $T' = (V, E' \cup \{e\} \setminus \{f\})$ הוא עץ פורש מינימלי, והוא מקיים את תנאי השמורה לאחר השלב ה־i.

שימוש בכלל האדום: נניח שצבענו את הקשת e באדום לפי המעגל הפשוט C. אם $e \notin E'$ אז מגדירים $T' = T$ ואין בעיה. אם $e \in E'$, אז נבדוק את החתך (U_e, W_e). לחתך הזה יש את הקשת e המשותפת עם המעגל C, ולכן חייבת להיות למעגל ולחתך לפחות עוד קשת אחת משותפת f (וזו אינה ב־E', כי יש רק קשת עץ אחת ב־(U_e, W_e) לפי הגדרת החתך). מתקיים $w(e) \geq w(f)$, כי e היא בעלת משקל מקסימלי במעגל לפי הגדרת הכלל האדום. בנוסף, לפי טענת המינימליות בחתך (U_e, W_e) מתקיים $w(f) \geq w(e)$, ז"א ששתי הקשתות מקבלות את המשקל המינימלי בחתך (U_e, W_e). על כן לפי אותה טענה גם העץ $T' = (V, E' \cup \{f\} \setminus \{e\})$ הוא עץ פורש מינימלי, והוא מקיים את תנאי השמורה לאחר השלב ה־i. בזאת כיסינו את שני המקרים והשלמנו את ההוכחה. ■

נסיים את הדיון כאן במסקנה קומבינטורית של הנכונות של האלגוריתם הגנרי.

משפט (אפיון של עצים פורשים מינימלים): עץ פורש $T = (V, E')$ הוא מינימלי אם ורק אם כל קשת $e \in E'$ היא בעלת משקל מינימלי בחתך (U_e, W_e), וזה קורה אם ורק אם כל קשת $e \in E \setminus E'$ היא בעל משקל מקסימלי במעגל C_e.

הוכחה: כיוון אחד במשפט (שהתנאים הנ"ל הם הכרחיים עבור עץ מינימלי) הוכחנו למעלה, בטענת המינימליות בחתך וטענת המקסימליות במעגל. נוכיח את הכיוון השני עבור כל אחד מהתנאים.

הכיוון השני עבור מינימליות בחתך: אם T מקיים שכל קשת $e \in E'$ היא בעלת משקל מינימלי בחתך (U_e, W_e), אז ראשית נפעיל את הכלל הכחול עבור כל קשת e ביחס לחתך (U_e, W_e). נקבל אז $E_b = E'$, נמשיך להריץ את האלגוריתם לפי הטענה על השלמת צביעה עד שכל שאר הקשתות יצבעו אדום, ונקבל את T כפלט האלגוריתם. ממשפט הנכונות הוא יהיה עץ פורש מינימלי כנדרש.

הכיוון השני עבור מקסימליות במעגל: אם T מקיים שכל קשת $e \notin E'$ היא בעלת משקל מקסימלי ב־C_e, אז ראשית נפעיל את הכלל האדום עבור כל קשת $e \in E \setminus E'$ ביחס למעגל C_e. נקבל אז $E_r = E \setminus E'$, ונמשיך להריץ את האלגוריתם עד שכל שאר הקשתות יצבעו כחול (הן לא יצבעו אחרת כי אנחנו יודעים שבסוף E_b הוא עץ פורש), ונקבל את T כפלט האלגוריתם. ממשפט הנכונות הוא יהיה עץ פורש מינימלי כנדרש. ■

אלגוריתמים ספציפים לעץ פורש מינימלי

נראה כאן שני אלגוריתמים ספציפים. בשניהם, עבור המימוש היעיל ביותר, יהיה צריך להשתמש במבני נתונים מתקדמים יחסית.

נתחיל מהאלגוריתם של קרוסקל Kruskal. באלגוריתם זה נחזיק בכל רגע נתון חלוקה של קבוצת הצמתים V – זאת בעצם תכיל את רשימת רכיבי הקשירות של (V, E_b), כאשר E_b היא קבוצת הקשתות שכבר קבענו להיות בעץ. באלגוריתם אנחנו עוברים על קשתות הגרף G בסדר מסויים, לא שרירותי. אח"כ נראה איך מממשים את אלו.

בתיאור האלגוריתם יש הערות על קשתות אדומות וכחולות, ולאחר כתיבתו נראה איך הצעדים באמת מתאימים ליישום של החוקים המתאימים באלגוריתם הגנרי.

האלגוריתם של קרוסקל

קלט: גרף קשיר ולא מכוון $G = (V, E)$, פונקצית משקל $w : E \to \mathbb{R}^+$

פלט: עץ פורש $T = (V, E_b)$ עם משקל מינימלי

- **אתחול:** $E_b \leftarrow \emptyset$, ומבנה החלוקה מאותחל לחלוקה לסינגלטונים $D \leftarrow \{\{v\} : v \in V\}$
- **עבור כל** קשת $uv \in E$, כאשר עוברים על הקשתות לפי **סדר לא־יורד** של ערכי $w(uv)$
  - תהי D_u הקבוצה בחלוקה D שמכילה את u, ו־D_v הקבוצה שמכילה את v
  - **אם** $D_u = D_v$
    - לא עושים כלום (קשת אדומה)
  - **אם** $D_u \neq D_v$
    - מוסיפים את uv ל־E_b (קשת כחולה)
    - מחליפים ב־D את הקבוצות D_u, D_v באיחוד שלהן $D_u \cup D_v$

לפני שנגיע למימוש ולזמן ריצה, ניגש להוכחת הנכונות.

טענה (רכיבי קשירות): בכל רגע נתון, D היא אכן רשימת רכיבי הקשירות של (V, E_b).

הוכחה: באינדוקציה על מספר הקשתות שעברנו עליהן. בבסיס, $\{\{v\} : v \in V\}$ היא אכן רשימת רכיבי הקשירות של הגרף חסר הקשתות $(V, \emptyset)$. עבור המעבר, המקום היחידי שבו מבצעים שינויים הוא המקרה $D_u \neq D_v$. אתם מוזמנים להראות שההוספה ל־E_b של הקשת uv בין שני רכיבי הקשירות אכן הופכת אותם לרכיב קשירות אחד, מבלי להשפיע על שאר הרשימה. ■

טענה (נכונות): האלגוריתם בפרט מיישם את האלגוריתם הגנרי, ולכן בסופו E_b הוא עץ פורש מינימלי.

הוכחה: ראשית, שימו לב שאנחנו עוברים על כל קשת פעם אחת בדיוק, ואז מייד "צובעים" אותה (מפעילים את אחד המקרים עבור הכרעה סופית האם היא תהיה בעץ או לא). נשתמש בטענה על רכיבי

הקשירות. במקרה ש־$D_u = D_v$, הדבר אומר שיש מסלול של קשתות כחולות בלבד בין u ל־v, ולכן ניתן להשתמש בכלל האדום (אין צורך להחזיק את הצביעה באופן מפורש, כי ממילא לא נחזור לבדוק את הקשת הזו).

במקרה ש־$D_u \neq D_v$, נגדיר את החתך $(D_u, V \setminus D_u)$. הקשת uv נמצאת בחתך (כי $D_v \subseteq V \setminus D_u$), וגם אנחנו יודעים שאין קשתות כחולות בחתך, בגלל ש־D_u רכיב קשירות של (V, E_b). כמו כן אנחנו יודעים שזו קשת לא־צבועה עם משקל מינימלי בחתך, כי אנחנו עוברים על הקשתות לפי סדר משקל לא־יורד, וכבר צבענו את כל קשתות הגרף שעברנו עליהן לפני uv. ■

עתה נפנה לדיון בזמן הריצה. על מנת לעבור על הקשתות לפי סדר משקל לא־עולה, עלינו קודם כל למיין אותן. הזמן (כזכור עבור מיון סדרת מספרים) יהיה $O(|E|\log(|E|)) = O(|E|\log(|V|))$. נותר לנו לבנות מבנה נתונים עבור D שגם יוכל לעשות את כל הפעולות בזמן כולל של $O(|E|\log(|V|))$ או פחות. מבנה נתונים של חלוקה שיודע לבצע את הפעולות של זיהוי קבוצה מכילה ושל איחוד קבוצות נקרא Union-Find. נראה כאן אחד פשוט שיהיה מספיק טוב לצרכים שלנו.

מבנה הנתונים: לכל $v \in V$ נחזיק מזהה קבוצה (בעת האיתחול יהיו לנו $|V|$ מזהים שונים). לכל קבוצה נשמור את האורך שלה וכן רשימה מקושרת של הצמתים החברים בה. כאשר מאחדים שתי קבוצות, צמתי הקבוצה הקטנה יותר יקבלו את מזהי הקבוצה הגדולה יותר (אם הגדלים זהים אז בוחרים שרירותית), והרשימות המקושרות ישורשרו בהתאם. שימו לב שמהגדרת המבנה נובע שפעולת חיפוש הזהות של קבוצה המכילה צומת נתון לוקחת זמן $O(1)$ (לכל צומת שמרנו מצביע ישיר לזהות הקבוצה שלו).

טענה (זמן ריצה כולל של מבנה הנתונים לתחזוק חלוקה): בעת ביצוע סדרה של $|V|-1$ איחודים בתוך המבנה (בלי קשר לסדר האיחודים) זמן הריצה הכולל הוא לא יותר מ־$O(|V|\log(|V|))$ (במקרה שלנו זה יבלע בזמן הריצה לשאר האלגוריתם של קרוסקל).

הוכחה: מרבית הזמן המושקע (זמן שאר הפעולות נבלע בזה) הוא זה של שינוי מזהי כל הצמתים בקבוצה אשר מאחדים אותה עם קבוצה לא קטנה ממנה. במקום לנתח לכל איחוד לחוד, עדיף לנתח לכל צומת את מספר הפעמים הכולל בכל האיחודים שהיה צריך לשנות את מזהה הקבוצה שלו (הסכום יהיה אותו דבר). כל פעם שצומת משנה מזהה, גודל הקבוצה שהוא חבר בה גדל לפחות פי 2 (משנים את מזהי הקבוצה הקטנה יותר), ולכן צומת לא יכול לעבור יותר מ־$O(\log(|V|))$ שינויים. לכן הסכום לכל הצמתים הוא $O(|V|\log(|V|))$ כנדרש. ■

כדאי להעיר שלפעמים קוראים לזמן ריצה עם ניתוח כמו בהוכחה האחרונה זמן ריצה ממוצע או מגולם amortized. במקרה שלנו יש זמן ריצה ממוצע של $O(\log(|V|))$ לכל פעולה. נעיר גם שידוע מבנה נתונים יותר מתוחכם של Union-Find, שבו זמן הריצה הכולל לפעולות הוא $O(|V|\alpha(|V|))$, כאשר $\alpha(k)$ היא פונקציה עולה מאוד אטית (בפרט אטית יותר מ־$\log(k)$, $\log\log(k)$, וכו).

עתה נעבור לדון באלגוריתם של פרים Prim. הרעיון כאן הוא להתרכז בקבוצת צמתים ולא בקשתות. בכל שלב נוסיף צומת חדש שיש לו קשת לצמתים הקודמים. אנחנו נבחר תמיד את השיטה ה"זולה" ביותר להוסיף צומת כזה. לפני שנדון במבנה הנתונים הדרוש למימוש מהיר, ניתן את החלק המתמטי של האלגוריתם באופן פורמלי.

האלגוריתם של פרים

קלט: גרף קשיר ולא מכוון $G = (V, E)$, פונקצית משקל $w : E \to \mathbb{R}^+$

פלט: עץ פורש $T=(V,E_b)$ עם משקל מינימלי

- **אתחול:** $E_b \leftarrow \emptyset$, $V' \leftarrow \{r\}$ (את r בוחרים שרירותית מ־V)
- **כל עוד** $V' \neq V$
  - תהי uv קשת קלה ביותר בין $u \in V \setminus V'$ כל שהוא ל־$v \in V'$ כל שהוא
  - מוסיפים את u ל־V' ואת uv ל־E_b (קשת כחולה, שאר הקשתות מ־u ל־V' נחשבות לאדומות)

טענה מיידית (קשירות): בכל שלב של האלגוריתם (V',E_b) הוא עץ פורש עבור V'. שיטה נחמדה להוכיח זאת היא לשים לב שיש כאן בפרט יישום של האלגוריתם הגנרי לחיפוש מ־r (שונה מחיפוש לרוחב ומחיפוש לעומק). ■

טענה (נכונות): האלגוריתם בפרט מיישם את האלגוריתם הגנרי למציאת עץ פורש מינימלי.

הוכחה: ההוספה של הקשת uv היא בדיוק הפעלת הכלל הכחול על $(V', V\setminus V')$. מכיוון ש־(V',E_b) קשיר, גם סימון הקשתות האדומות נכון. ■

על מנת לנתח זמן ריצה, צריך לנסח את האלגוריתם עם מבני הנתונים הדרושים.

האלגוריתם של פרים עם פירוט הנתונים המתוחזקים בהרצה

קלט: גרף קשיר ולא מכוון $G=(V,E)$, פונקצית משקל $w: E\rightarrow \mathbb{R}^+$

פלט: עץ פורש $T=(V,E_b)$ עם משקל מינימלי

- **אתחול:** $E_b \leftarrow \emptyset$, $Q \leftarrow V$ ולכל $v \in V$ מציבים $k(v) \leftarrow +\infty$ ו־$l(v) \leftarrow \mathrm{nil}$
(כאן Q הוא $V \setminus V'$, הפנקציה k תתן לכל $v \in Q$ את המשקל המינימלי של קשת ממנו ל־V', ו־l תחזיק מצביע לצומת השני של קשת מינימלית כזו)
- **עבור** r אחד **שרירותי** מציבים $k(r) \leftarrow 0$ (הוא יכנס ראשון ל־V')
- **כל עוד** $Q \neq \emptyset$
  - יהי u צומת מתוך Q עבורו $k(u)$ מינימלי
  - מורידים את את u מ־Q, ואם $l(u) \neq \mathrm{nil}$ מוסיפים את $u, l(u)$ ל־E_b
  - **לכל** שכן v של u, אם $k(v) > w(uv)$
    * מציבים $k(v) \leftarrow w(uv)$ ו־$l(v) \leftarrow u$

טענה (נכונות): בכל רגע נתון לאחר הכנסת r, לכל צומת $u \in Q$ הערך $k(u)$ הוא משקל של קשת קטנה ביותר בין u ל־$V' = V \setminus Q$, ו־$+\infty$ אם אין קשת ל־V', בעוד $l(u)$ מצביע לצומת השני בקשת כזו אם היא קיימת. על כן, האלגוריתם הזה מיישם את האלגוריתם של פרים.

הוכחה: ההוכחה היא באינדוקציה על מספר הצעדים. הבסיס הוא מייד לאחר הוצאת הצומת הראשון r (אז השכנים של r מתעדכנים בקשת שלהם אליו). לא קשה לראות שלולאת העדכון של השכנים לאחר הוצאת צומת מ־Q משאירה את השמורה הנ"ל על ערכי k ו־l. ■

למימוש מבנה הנתונים המחזיק את Q ואת ערכי k לצמתי Q (אין צורך לשמור אותם לצמתים שאינם ב־Q), הדרך הפשוטה והנאיבית ביותר היא ע"י מערכים. בכל שלב, מציאת הצומת עם הערך המינימלי לוקחת $O(|V|)$, העדכון בהוצאת צומת u מ־Q לוקח זמן $O(\deg(u))$, ובסוף $|V|$ איטרציות נקבל זמן כולל $O(|V|^2 + \sum_{u \in V} \deg(u)) = O(|V|^2)$.

נקבל אבל זמני ריצה טובים בהרבה אם במקום זאת נחזיק את ערכי k במבנה הנתונים המוכר של ערימה heap. פעולת הוצאת איבר מינימלי תיקח $O(\log(|V|))$, ופעולת עדכון ערך בודד גם תיקח $O(\log(|V|))$. זמן הריצה הכולל יקבע ע"י סך פעולות העדכון (שאר הפעולות יקחו פחות זמן), שזה $O(\sum_{u \in V} \deg(\mathrm{u}) \cdot \log(|\mathrm{V}|)) = \mathrm{O}(|\mathrm{E}| \log(|\mathrm{V}|))$. זמן האכלוס של הערימה הוא $O(|V|)$ בלבד, כי בהתחלה כל ערכי k הם $+\infty$ פרט לאחד מהם.

נעיר שיש מבנה נתונים יותר מתקדם, ערימת פיבונאצ'י, שמבטיח שפעולות הקטנת ערכי k יקחו זמן ממוצע (amortized) של $O(1)$ לכל פעולה (אין באלגוריתם פעולות הגדלה), ואז נוכל "לסחוט" מהאלגוריתם זמן כולל של $O(|E| + |V| \log(|V|))$ (המחובר השני הוא כי כל פעולת הוצאה עדיין לוקחת $O(\log(|V|))$ זמן ריצה). נעיר לסיום שכיום ידוע אלגוריתם לעץ פורש מינימלי בזמן $O(|E|\alpha(|E|))$ (כאשר $\alpha(k)$ היא הפונקציה האיטית שמופיעה גם במבנה המתוחכם יותר עבור Union-Find), והשאלה האם קיים אלגוריתם שרץ בזמן $O(|E|)$ היא עדיין פתוחה.

אלגוריתמים חמדניים

אלגוריתמים חמדניים greedy algorithms הם אלגוריתמים שעובדים לפי העיקרון הכללי של בניית פיתרון בשלבים, כשבכל שלב לוקחים את האופציה ה"טובה ביותר לאותו רגע". האלגוריתם של קרוסקל ("כל פעם לוקחים את הקשת המינימלית שמאחדת רכיבי קשירות") ושל פרים ("כל פעם לוקחים קשת מינימלית שמרחיבה את V'") הם דוגמאות לאלגוריתמים חמדניים. בדרך כלל החלק היותר קשה הוא להוכיח שאלגוריתם חמדני באמת יתן פיתרון מיטבי, או קירוב של פיתרון כזה. כמובן שיש גם מקרים שבהם אלגוריתמים חמדניים לא יעבדו טוב.

בפרק זה נראה דוגמאות נוספות לניתוח של אלגוריתמים חמדניים שנותנים פיתרון מיטבי.

שתי בעיות שיבוץ ב"ציר זמן"

הבעיה הראשונה שלנו תהיה של מציאת מספר מקסימלי של קטעים זרים בזוגות. הקלט יהיה קבוצה של קטעים על הישר (נתעלם כרגע מהדיון על ייצוג מספרים מהישר). באופן פורמלי, $S = \{a_1, \ldots a_n\}$ כאשר $a_i = (s_i, f_i)$. אפשר לחשוב על כל קטע כנתון ע"י "זמן התחלה" ו"זמן סיום". למשל, זמני הרצאות

מעניינות בפקולטה שהייתם רוצים לשמוע כמה שיותר מהן. הפלט צריך להיות תת־קבוצה $S' \subseteq S$, כך שכל הקטעים ב־S' זרים זה לזה ו־$|S'|$ הוא המקסימלי האפשרי בין קבוצות אלו.

השיטה החמדנית שלנו תהיה להתחיל מ־$S' = \emptyset$, וכל פעם להוסיף ל־S' קטע זר לקטעים הקודמים אשר מסתיים הכי מוקדם שאפשר. החמדנות מתבטאת בכך שכל פעם רוצים "להתפנות הכי מוקדם שאפשר". נניח כאן שהמדובר בקטעים פתוחים, ז"א שמותר שקטע ב־S' יתחיל בדיוק בנקודה שבה הקטע הקודם נגמר (לא קשה לשנות את האלגוריתם למקרה שלא מרשים כזה דבר).

אלגוריתם חמדני (כולל מימוש) לקבוצת קטעים זרים מקסימלית

קלט: קבוצת קטעים S

פלט: קבוצה $S' \subseteq S$ בגודל מקסימלי שמכילה קטעים זרים זה לזה

- **אתחול:** ממיינים את S לפי סדר לא־יורד של זמני סיום $f_1, \ldots, f_n$, ומציבים $S' \leftarrow \emptyset$, $f \leftarrow -\infty$ (בכל רגע נתון f יחזיק את זמן הסיום של הקטע האחרון שהוכנס ל־S')
- **עבור כל** קטע $(s_i, f_i) \in S$, **לפי סדר** f_i לא־יורד
  - **אם** $s_i \geq f$
    - מוסיפים את (s_i, f_i) ל־S'
    - מציבים $f \leftarrow f_i$

זמן הריצה של האלגוריתם (בהנחה שלוקח זמן $O(1)$ להשוות שני מספרים) הוא בבירור $O(n \log(n))$, הזמן שלוקח למיין את S בעת האתחול.

משפט (נכונות): הקבוצה המוחזרת S' היא אכן קבוצה עם מספר מקסימלי של קטעים זרים (יכול להיות שהיא לא הקבוצה היחידה עם תכונה זו).

הוכחה: נראה באינדוקציה על i את הטענה שלאחר המעבר על הקטע (s_i, f_i), קיימת קבוצת קטעים זרים מגודל מקסימלי $\hat{S}$ שעבורה $S' \cap \{a_1, \ldots, a_i\} = \hat{S} \cap \{a_1, \ldots, a_i\}$. משפט הנכונות נובע מטענה זו עבור $i = n$ (נסו לראות כאן קווי דמיון להוכחה של האלגוריתם הגנרי עבור עצים פורשים מינימלים).

הבסיס הוא $i = 0$, שאז המשוואה של הטענה נהיית "$\emptyset = \emptyset$".

עבור צעד האינדוקציה, נניח שקיימת קבוצה $\hat{S}$ שעבורה $S' \cap \{a_1, \ldots, a_{i-1}\} = \hat{S} \cap \{a_1, \ldots, a_{i-1}\}$ וננתח מה קורה כשהאלגוריתם בודק את הקטע $a_i = (s_i, f_i)$.

אם a_i לא הוכנס ל־S', אז הוא לא זר ל־$S' \cap \{a_1, \ldots, a_{i-1}\}$, ולכן הוא גם לא יכול להיות ב־$\hat{S}$, ובפרט $S' \cap \{a_1, \ldots, a_i\} = \hat{S} \cap \{a_1, \ldots, a_i\}$.

אם a_i הוכנס ל־S' והוא נמצא ב־$\hat{S}$, אז גם $S' \cap \{a_1, \ldots, a_i\} = \hat{S} \cap \{a_1, \ldots, a_i\}$.

במקרה שנותר לנתח, יהי $j > i$ האינדכס המינימלי אחרי $i - 1$ כך ש־$a_j \in \hat{S}$. אינדכס כזה קיים כי אחרת היה אפשר להוסיף את a_i ל־$\hat{S}$ ולקבל סתירה למקסימליות שלה. נסתכל עתה על $\hat{S} \cup \{a_i\} \setminus \{a_j\}$.

הקבוצה הזו היא באותו גודל כמו $\hat{S}$, וגם היא מורכבת מקטעים זרים: זאת מכיוון ש־$f_i \leq f_j$ (עברנו על הקטעים לפי סדר f_i לא־יורד), ולכן ב־$\hat{S}$ אין קטעים עם אינדכס גדול מ־j שאינם זרים ל־a_i (קטעים כאלו גם היו לא־זרים ל־a_j, בגלל שזמן הסיום שלהם היה לפחות f_j).

בזאת כיסינו את כל המקרים האפשריים עבור צעד האינדוקציה. ■

עתה נעבור לבעיה השניה, מעין בעיית־אריזה. נניח שיש לנו משימות $a_1, \ldots, a_n$, כאשר לכל משימה a_i נתון משך הזמן הדרוש לביצוע t_i, וכן זמן סיום רצוי d_i. אפשר לחשוב על קבלן שקיבל על עצמו משימות (אולי יותר מדי מהן?) עם מועדי סיום מוגדרים. אנחנו צריכים למצוא סידור של המשימות שיהיה לו "זמן איחור מקסימלי" קטן ביותר.

יותר פורמלית, סידור אפשרי ניתן ע"י פרמוטציה $\sigma \in S_n$. זה אומר שלכל i נסיים את המשימה $a_{\sigma(i)}$ בזמן $f_{\sigma(i)} = \sum_{j=1}^{i} t_{\sigma(j)}$, ובהתאם האיחור שלה יהיה $f_{\sigma(i)} - d_{\sigma(i)}$. "איחור" יכול להיות שלילי אם השלמנו את המשימה לפני הזמן. האיחור המקסימלי הוא $L(\sigma) = \max_{1\leq j\leq n}(f_j - d_j) = \max_{1\leq i\leq n}(f_{\sigma(i)} - d_{\sigma(i)})$. נרצה למצוא σ שעבורו ערך זה הוא המינימלי ביותר האפשרי.

העיקרון החמדני שלנו יהיה פשוט ביותר: נבצע בכל פעם את המשימה עם זמן הסיום המוקדם ביותר, או במלים אחרות נעבור על המשימות לפי סדר זמני סיום לא־יורד.

"אלגוריתם" חמדני לסידור עם מינימום של זמן איחור מקסימלי

קלט: סדרת משימות $a_1, \ldots, a_n$ עם זמני ביצוע $t_1, \ldots, t_n$ וזמני סיום רצוי $d_1, \ldots, d_n$, בהתאמה

פלט: פרמוטציה $\sigma \in S_n$ שמשיגה את המינימום עבור זמן איחור מקסימלי $L(\sigma)$

- מגדירים את σ להיות הפרמוטציה כך ש־$d_{\sigma(1)}, \ldots, d_{\sigma(n)}$ היא התוצאה של המיון של $d_1, \ldots, d_n$ בסדר לא־יורד.

זמן הריצה הוא גם כאן $O(n \log(n))$ מהמיון. על מנת להראות נכונות מראים את למת ההחלפה הבאה.

למה (חילוף בודד): אם σ היא פרמוטציה כזו שעבור j מסויים $d_{\sigma(j)} \geq d_{\sigma(j+1)}$, ו־$\sigma'$ מתקבלת מ־σ ע"י קביעת $\sigma'(j+1) = \sigma(j)$ ו־$\sigma'(j) = \sigma(j+1)$ (בעוד על שאר הערכים שתי הפרמוטציות מסכימות), אז $L(\sigma') \leq L(\sigma)$.

הוכחה: נסמן ב־$f'_1, \ldots, f'_n$ את זמני הסיום עבור σ'. לכל $k \in \{1, \ldots, n\} \setminus \{j, j+1\}$ לא קשה לראות שמתקיים $f'_{\sigma'(k)} = f_{\sigma(k)}$, ושאלו ישחקו אותו תפקיד בחישוב המקסימום עבור $L(\sigma)$ ו־$L(\sigma')$. יש עוד שני איברים בחישוב ערכי המקסימום הנ"ל. כדאי לשים לב שגם מתקיים $f'_{\sigma'(j+1)} = f_{\sigma(j+1)}$ (אבל שם ערך ה־d אינו בהכרח זהה). לסיום נראה שגם $f'_{\sigma'(j)} - d_{\sigma'(j)}$ וגם $f'_{\sigma'(j+1)} - d_{\sigma'(j+1)}$ אינם עולים על $f_{\sigma(j+1)} - d_{\sigma(j+1)}$, וזה יספיק עבור טענת המקסימום.

עבור ההפרש הראשון: $f'_{\sigma'(j)} - d_{\sigma'(j)} \leq f'_{\sigma'(j+1)} - d_{\sigma'(j)} = f_{\sigma(j+1)} - d_{\sigma(j+1)}$ (המעבר הראשון הוא ממונוטוניות ערכי f', והשני מההגדרה של σ' ומהדיון למעלה).

עבור ההפרש השני: $f'_{\sigma'(j+1)} - d_{\sigma'(j+1)} = f_{\sigma(j+1)} - d_{\sigma(j)} \leq f_{\sigma(j+1)} - d_{\sigma(j+1)}$ (המעבר הראשון הוא לפי ההגדרות והדיון למעלה, והמעבר השני לפי ההנחה $d_{\sigma(j)} \geq d_{\sigma(j+1)}$).

ביחד קיבלנו את הטענה המבוקשת. ■

הסיבה שבניסוח הלמה כתבנו "$\geq$" במקום רק "$>$" היא כדי שנוכל להראות שכל סידור לא־יורד של $d_1, \ldots, d_n$ עובד, ולא רק סידור אחד ספציפי. נראה את זה פורמלית.

משפט (נכונות): אם $d_{\sigma(1)}, \ldots, d_{\sigma(n)}$ מסודרים בסדר לא־יורד, אז $L(\sigma)$ מקבל ערך מינימלי.

הוכחה: נניח ש־π מקבלת את ערך המינימום $L(\pi)$, ונגדיר סדרת פרמוטציות $\sigma_0, \ldots \sigma_r$ כך ש־$\sigma_0 = \pi$, $\sigma_r = \sigma$, ולכל i מקבלים את σ_i מתוך σ_{i-1} ע"י חילוף בודד של $\sigma_{i-1}(j), \sigma_{i-1}(j+1)$ כאשר הערכים שם מקיימים $d_{\sigma_{i-1}(j)} \geq d_{\sigma_{i-1}(j+1)}$. לפי הלמה על חילוף בודד נובע מכך $L(\sigma_i) \leq L(\sigma_{i-1})$, ולבסוף $L(\sigma_r) \leq L(\sigma_0)$ כנדרש.

על מנת לייצר את סדרת הפרמוטציות, עושים מיון בועות bubble sort על $d_{\pi(1)}, \ldots, d_{\pi(n)}$. על מנת "להכריע" בין שני ערכי d שווים, מגדירים ש־d_k צריך להיות לפני d_l אם $d_k < d_l$ (כרגיל), או אם $d_k = d_l$ וגם $\sigma^{-1}(k) < \sigma^{-1}(l)$ (או במילים אחרות σ שמה את k לפני l). התוצאה של סוף המיון הזה תהיה בדיוק σ, וכמו בכל מיון בועות, כל הפעולות הן פעולות של חילוף שני איברים עוקבים בסידור הנוכחי, שעבורם ערכי d הם בסדר הלא־נכון (או שערכי d שווים וערכי σ^{-1} מסודרים לא־נכון). כל שנותר לעשות הוא להגדיר את ה־σ_i לפי כל סידורי הביניים שאלגוריתם מיון הבועות עובר דרכם. ■

עצי האפמן עבור קודים חסרי רישות

נתמקד עתה בשאלה של דחיסת מידע. נתון א"ב סופי $\Sigma = \{s_1, \ldots, s_n\}$. קידוד שלו הוא התאמה של מילה בינארית סופית לכל אות, $C : \Sigma \to \{0,1\}^*$. באופן טבעי נרחיב את זה לקידוד של מילים מעל Σ ע"י שרשור: $C(t_1, \ldots, t_k) = C(t_1)C(t_2)\cdots C(t_k)$. באופן טבעי נרצה קוד חד־פענח, או במילים אחרות, נדרוש שההרחבה $C : \Sigma^* \to \{0,1\}^*$ היא חח"ע (חד־חד־ערכית).

אפשרות אחת היא שכל מילות הקוד יהיו בנות אותו מספר ביטים, כמו שנעשה ב־ASCII (הקוד ההסטורי לייצוג אותיות אנגליות וסימנים בסיסיים במסמך טקסט ממוחשב). משפחה יותר רחבה של קודים שהם חד־פענח היא זו של קודים חסרי רישות prefix codes. קוד ייקרא חסר רישות אם אין אף $a, b \in \Sigma$ (שונים זה מזה) שעבורם $C(a)$ הוא רישה של $C(b)$.

המשפט הבא, שמובא כאן ללא הוכחה, מראה שאנחנו לא צריכים לטרוח במציאת קודים חד־פענח שאינם חסרי רישות.

משפט (נובע מאי־שוויון קראפט): לכל קוד חד־פענח C קיים קוד חסר רישות C' כך שלכל $a \in \Sigma$ מתקיים $|C(a)| = |C'(a)|$.

כאשר אנחנו יודעים מראש כמה מופעים יהיו מכל אות, נוכל לבנות קוד חסר רישות שייקח פחות ביטים לאחסון מאשר קוד עם אורך קבוע. נוכל לדאוג שלאותיות שמופיעות יותר פעמים יהיו קידודים יותר קצרים. הבעיה שנרצה לפתור היא מציאת קידוד, בהינתן מספר המופעים של כל אות, שייתן את האורך המינימלי למילת הקוד.

דוגמה: נניח שנרצה לקודד את המחרוזת "גנן גדל דגן בגן". בדוגמה שלנו נניח ש־"נ" ו־"ן" זו אותה אות, ושרווח הוא גם אות שצריך לקודד. קוד אורך קבוע יצריך 3 ביטים לכל אות, וזה יתן סה"כ 45 ביטים בקידוד. אבל, נניח שניתן את הקידוד הבא:

ל	ב	ד	רווח	ג	נ
1111	1110	110	10	01	00

אז נקבל 36 ביטים בקידוד בלבד. בקרוב נראה איך היינו יכולים למצוא את הקידוד הזה אלגוריתמית.

לפני שנמשיך, כדאי לראות קשר בין קודים חסרי רישות לבין עצים בינארים. עץ בינארי הוא עץ עם שורש שבו לכל צומת יש לכל היותר שני בנים – לכל היותר בן אחד עם תווית "0", ולכל היותר בן אחד עם תווית "1" (התוויות בעצם מיוחסות לקשתות מאב לבן). אם יש לנו עץ בינארי T בתוספת התאמה חח"ע ועל בין הא"ב Σ לבין קבוצת העלים L של העץ, אז אפשר לבנות קוד חסר רישות באופן הבא: לכל $a \in \Sigma$, יהי v_a העלה של העץ המותאם אליו. הקידוד $C_T(a)$ יוגדר כמחרוזת התוויות המופיעות לפי הסדר במסלול מהשורש של T לעלה v_a.

טענה: עבור עץ T עם תוויות כפי שמתואר למעלה הקוד C_T הוא חסר רישות.

סקיצת הוכחה: עבור כל צומת v (עלה או לא) ב־T נגדיר את w_v להיות המחרוזת של התויות על הקשתות במסלול מהשורש של T ל־v. אפשר לראות באינדוקציה על העומק (המרחק מהשורש) של v שאורך המילה $|w_v|$ הוא עומק הצומת, וכן שאין שני צמתים $u \neq v$ עבורם $w_u = w_v$. מכאן אפשר לראות שאם w_u הוא רישה של w_v אז u הוא אב קדמון של v (כי האב הקדמון יקבל את מילת הרישה, והרגע ראינו שרק צומת אחד יכול לקבל אותה). לסיום, משתמשים בזה שבהגדרות למעלה רק עלים מקבלים תוויות מ־Σ, בעוד שהרישות של המילים המתאימות לעלים כולן מתאימות לצמתים פנימיים. ■

עתה נראה בכיוון ההפוך, שלכל קוד חסר רישות יהיה עץ מתאים. אם $C : \Sigma \to \{0,1\}^*$ הוא קוד חסר רישות, אז נגדיר את קבוצת הצמתים V להיות תת־קבוצה סופית של $\{0,1\}^*$. במדויק, זו תהיה קבוצת כל המילים שהן רישות של איברים בתמונת הקוד $\{C(a) : a \in \Sigma\}$. שימו לב שבפרט המילה הריקה תמיד תהיה אחד האיברים ב־V. היא תהיה שורש העץ שלנו. הקשתות עם תווית "0" יהיו איברי הקבוצה $\{uv : u, v \in V,\, v = u0\}$ (ז"א זוגות המילים מ־V כשהשניה מתקבלת מהראשונה ע"י שרשור של התו "0" בסוף), והקשתות עם תווית 1 יהיו איברי הקבוצה $\{uv : u, v \in V,\, v = u1\}$. לכל איבר $a \in \Sigma$ נתאים את הצומת $C(a) \in V$.

טענה: הגרף המתקבל הוא באמת עץ.

הוכחה: על מנת להראות קשירות, אפשר לראות שיש מסלול מהשורש לכל מילה u שמתאימה לצומת בגרף. זהו המסלול העובר דרך כל הרישות של u (לפי הבניה שלנו, לכל מילה שהכנסנו לקבוצת הצמתים הכנסנו גם את כל הרישות שלה). על מנת להראות חוסר מעגלים, אם היה מעגל, אז נסמן בו את אחד הצמתים המתאימים למילה מאורך מקסימלי ב־u, ונסתכל על השכנים שלו במעגל w ו־v. אלו מתאימים לשתי רישות מאורך $|u| - 1$ שאמורות להיות שונות זו מזו, ודבר זה לא ייתכן. ■

לסיום, אתם מוזמנים לנסות להוכיח את הטענה הבאה.

טענה עם הוכחה כתרגיל: בהגדרות למעלה נוצרת התאמה חח"ע ועל בין Σ וקבוצת העלים של העץ. יתרה מזו, אם נסמן את העץ המתקבל ב־T, אז C_T הוא בדיוק C שלפיו העץ הוגדר.

עבור עץ T, נסמן ב־$d_T(v)$ את עומק הצומת (כאשר שורש העץ הוא מעומק 0). עבור $a \in \Sigma$ יתקיים בפרט שהעומק של v_a שווה לאורך מילת הקוד, $d_T(v_a) = |C_T(a)|$. בהינתן פונקציה $f : \Sigma \to \mathbb{N}$ אשר מתארת את מספרי המופעים של כל אות במחרוזת נתונה w, אז מספר הביטים הכולל בקידוד של w לפי T יהיה $B(T) = \sum_{a\in\Sigma} d_T(v_a) f(a)$. עץ בעל ערך מינימלי לסכום $B(T)$ יקרא עץ האפמן Huffman.

הערה: ההגדרה של עץ האפמן הגיונית גם עבור פונקציה f בעלת ערכים ממשיים, למשל פונקציה שמתארת לכל $a \in \Sigma$ את "אחוז המופעים הצפוי" בטקסט שצריך יהיה לקודד.

עתה נוכיח דברים שעץ האפמן צריך לקיים. בכל הטענות נניח שכל הערכים של f גדולים מ־0 (פשוט לא נקודד איברים שלא מופיעים כלל במילה המקודדת).

טענה (מלאות): כל עץ האפמן הוא מלא – לכל צומת שאינו עלה יש שני בנים בדיוק.

הוכחה: נניח ש־u הוא צומת בעץ שיש לו בן אחד בדיוק, ונסמן את הבן ב־v. אם v הוא עלה, אז פשוט נסיר אותו ונעביר את התווית (מתוך Σ) שלו לצומת u, שנהיה עלה במקומו. מכיוון שלאף עלה לא הגדלנו את העומק, בעוד שהקטנו את העומק של העלה המתאים לאחת התוויות מ־Σ, הסכום $\sum_{a\in\Sigma}d_T(v_a)f(a)$ נהיה קטן יותר, ז"א שהעץ המקורי לא היה אופטימלי.

אם v אינו עלה, נסיר את v, ואת הבנים של v נהפוך לבנים של u (תוך כדי שמירת על התוויות של הקשתות מ־v לבנים המקוריים שלו). עתה לכל העלים שהם הצאצאים המקוריים של v (וחייב להיות לפחות עלה אחד כזה), העומק שלהם קטן ב־1, בעוד שלעלים שאינם צאצאים מקוריים של v העומק לא השתנה. שוב הגענו למצב שלאף עלה לא הגדלנו את העומק, בעוד שלפחות לעלה אחד הקטנו את העומק, ז"א שהעץ המקורי לא היה אופטימלי. ■

למה (מיקום לפי תדירות): אם $b,c\in\Sigma$ שני איברים בעלי תדירות מינימלית (לפי f), אז קיים עץ האפמן שבו v_b ו־v_c הם אחים ובעלי עומק מקסימלי.

הוכחה: נניח ש־T הוא עץ האפמן כל שהוא, וממנו נבנה עץ האפמן T' המקיים את המבוקש. קודם כל, לפי הטענה על מלאות קיימים לפחות שני עלים אחים בעלי עומק מקסימלי (כי אם u הוא עלה כזה, אז יש לו אח בגלל שהעץ מלא, והאח גם יהיה עלה כזה). נסמן את העלים הנ"ל ב־u ו־w. אם x ו־y הם האיברים עבורם $v_x=u$ ו־$v_y=w$, אז נניח שמתקיים $f(x)\leq f(y)$. כמו כן נניח שמתקיים $f(b)\leq f(c)$. מההנחות מתקיים $f(b)\leq f(x)$ ו־$f(c)\leq f(y)$, בגלל ש־b ו־c הם האיבר הכי נדיר והאיבר השני הכי נדיר (או איבר הכי נדיר נוסף).

עתה נבנה את T' ע"י כך שנחליף בין העלים v_x ו־v_b (אלא אם כן זהו כבר אותו עלה), ובין v_y ו־v_c (אלא אם כן זהו אותו עלה). במקרה ש־$v_b=v_y$ נחליף במקום זאת את v_x ו־v_c. נותר לשים לב שבהחלפות כאלו הסכום $B(T)=\sum_{a\in\Sigma}d_T(v_a)f(a)$ אינו גדל. למשל, בהחלפת העלים v_x ו־v_b ההפרש המתקבל $B(T')-B(T)$ הוא:

$$d_T(v_b)f(x)+d_T(v_x)f(b)-d_T(v_b)f(b)-d_T(v_x)f(x)=(f(x)-f(b))(d_T(v_b)-d_T(v_x))\leq 0$$

(השתמשנו כאן גם ב־$d_T(v_b)\leq d_T(v_x)$, לפי הנחת העומק על $u=v_x$). בשאר ההחלפות מגיעים למסקנה דומה, ז"א שהסכום המתאים לעץ החדש לא־עולה על זה של הישן. ■

הלמה האחרונה מרמזת על הרעיון הבא לבניית עץ האפמן באופן חמדני: בכל פעם ניקח שני איברים בעלי תדירות מינימלית, ובאופן זמני "נאחד" אותם. לאחר בניית עץ האפמן לקלט החדש המתקבל, נחליף את העלה המתאים לאותו איחוד בצומת ולו שני הבנים המתאימים. הלמה הבאה מתארת את הרעיון הזה באופן פורמלי.

למה (איחוד איברים נדירים): נניח ש־b ו־c הם שני איברים נדירים ביותר (לפי f) ב־Σ. נגדיר את $\Sigma'=\Sigma\setminus\{b,c\}\cup\{z_{bc}\}$, כאשר z_{bc} הוא איבר חדש לחלוטין, וכן נגדיר את $f':\Sigma'\to\mathbb{N}$ לפי $f'(z_{bc})=f(b)+f(c)$, כאשר לכל $a\in\Sigma'\setminus\{z_{bc}\}$ מתקיים $f'(a)=f(a)$.

נניח ש־T' הוא עץ האפמן של Σ' ו־f', ונגדיר את T ע"י כך שנחליף את העלה $v_{z_{bc}}$ בצומת פנימי ולו בן אחד v_b ובן אחד v_c, שניהם עלים חדשים (לא משנה איזו מהקשתות אליהם תקבל תווית "0" ואיזו "1"). העץ T יהיה עץ האפמן עבור Σ ו־f.

הוכחה: ראשית נניח ש־$\hat{T}$ הוא עץ האפמן עבור Σ ו־f שבו v_b ו־v_c הם אחים (קיים כזה לפי הלמה על מיקום לפי תדירות). נניח ש־w הוא האב המשותף של v_b ו־v_c. נבנה עץ $\hat{T}'$ עבור Σ' ע"י זה שנחליף את w ושני בניו בצומת חדש יחיד $v_{z_{bc}}$. נשים לב שמתקיים $d_{\hat{T}}(v_b) = d_{\hat{T}}(v_c) = d_{\hat{T}'}(v_{z_{bc}}) + 1$, ולכן מתקיים $d_{\hat{T}}(v_b)f(b) + d_{\hat{T}}(v_c)f(c) = d_{\hat{T}'}(v_{z_{bc}})f'(z_{bc}) + f(b) + f(c)$. נציב את זה בנוסחה ל־$B(\hat{T})$ ונקבל $B(\hat{T}) = \sum_{a\in\Sigma} d_{\hat{T}}(v_a)f(a) = \sum_{a\in\Sigma'} d_{\hat{T}'}(v_a)f'(a) + f(b) + f(c) = B(\hat{T}') + f(b) + f(c)$. כמו כן מההנחה על T', מתקיים $B(\hat{T}') \geq B(T')$. נשים לב שבאופן דומה לטיעון על $B(\hat{T})$ מתקיים $B(T) = B(T') + f(b) + f(c)$, ומכל אלו יחד נקבל $B(\hat{T}) \geq B(T)$. הנחנו ש־$\hat{T}$ הוא עץ האפמן, ולכן זהו שוויון ו־T הוא עץ האפמן גם. ■

לסיום נגסח אלגוריתם רקורסיבי עבור מציאת עץ האפמן.

אלגוריתם רקורסיבי למציאת עץ האפמן

קלט: אלפבית Σ מגודל לפחות 2, פונקצית שכיחויות $f:\Sigma\to\mathbb{R}^+$

פלט: עץ האפמן T

- **אם** $|\Sigma| = 2$ אז מחזירים ב־T עץ שבו שני איברי Σ מותאמים לשני בנים עלים של השורש.
- **אחרת** יהיו $b, c \in \Sigma$ שני האיברים עם ערכי f מינימלים
  - מגדירים את $\Sigma' = \Sigma \setminus \{b, c\} \cup z_{bc}$, ואת $f' : \Sigma' \to \mathbb{Z}$ לפי $f'(z_{bc}) = f(b) + f(c)$
  - **קוראים** לאלגוריתם באופן **רקורסיבי** עם Σ' ו־f' לקבלת עץ T'
  - מחליפים ב־T' את העלה $v_{z_{bc}}$ בצומת פנימי עם בנים עלים v_b ו־v_c, לקבלת T

טענה (נכונות): האלגוריתם יחזיר עץ האפמן עבור Σ ו־f.

הוכחה: באינדוקציה פשוטה על $|\Sigma|$. הבסיס ברור (עבור $\Sigma = \{a, b\}$ צריך ביט בודד לכל אחת מהאותיות, כי קידוד שמשתמש במילה הריקה אינו חסר־רישות). צעד המעבר הוא תוצאה ישירה של הלמה על איחוד איברים נדירים. ■

מימוש וזמן ריצה: אנחנו צריכים תמיד למצוא את שני האיברים עם ערכי f מינימלים. לשם כך נחזיק את כל איברי Σ בערימה heap ביחס לערכי f, שתישמר כמבנה נתונים גלובלי שנאתחל לפני הקריאה הראשונה לאלגוריתם הרקורסיבי. האכלוס של הערימה בעת האתחול לוקח זמן $O(|\Sigma|\log(|\Sigma|))$. נשים לב שהקריאה הרקורסיבית מתבצעת בדיוק פעם אחת ולאחריה אין יותר צורך בערכי f הקודמים, ולכן אפשר לבצע פעולות בלתי־הפיכות במבנה הנתונים כהכנה אליה.

לקראת הקריאה הרקורסיבית, נוציא מהערימה (בזה אחר זה) את שני האיברים בעלי הערך המינימלי, בזמן $O(\log(|\Sigma|))$, ונכניס לערימה את האיבר החדש z_{bc} עם ערכו $f'(z_{bc}) = f(b) + f(c)$, גם בזמן $O(\log(|\Sigma|))$. סה"כ מספר הקריאות הרקורסיביות הוא $|\Sigma| - 1$ (קריאה אחת לכל גודל א"ב בין 2 ל־$|\Sigma|$ המקורי), ולכן זמן הריצה הכולל שנקבל יהיה $O(|\Sigma|\log(|\Sigma|))$.

לסיום, כדאי לראות דוגמת הרצה של האלגוריתם המדובר. נחזור למחרוזת "גנן גדל דגן בגן". ערכי f נתונים ע"י הטבלא הבאה.

ל	ב	ד	רווח	ג	נ
1	1	2	3	4	4

הקריאה הראשונה תאחד את "ל" ואת "ב" לקראת הקריאה הרקורסיבית, שתתבצע עם הטבלא הבאה.

ל+ב	ד	רווח	ג	נ
2	2	3	4	4

האיחוד הבא יהיה בין "ל+ב" ו־"ד" לאיבר עם 4 מופעים, ולאחריו "ל+ב+ד" יאוחד עם הרווח. בסוף "ג" ו־"נ" יאוחדו. צעד הבסיס יבוצע עבור הא"ב הכולל רק את "ל+ב+ד+רווח" ואת "ג+נ", ולאחר שנבצע את ההחלפות בעץ לפי האיחודים שבוצעו נקבל את הקוד שהוצג קודם בהרצאה.

ל+ב+ד	רווח	ג	נ
4	3	4	4

ל+ב+ד+רווח	ג	נ
7	4	4

ל+ב+ד+רווח	ג+נ
7	8

מסלולים קלים ביותר / קצרים ביותר

הערה על המינוח: המונח המקובל באנגלית הוא shortest paths, "מסלולים קצרים ביותר". אנחנו נשתמש כאן במונח "מסלולים קלים ביותר", מכיוון שמרשים גם "אורכים" שליליים.

בפרק זה הגרף $G=(V,E)$ הוא פשוט, ובד"כ הוא יהיה מכוון. בנוסף תהיה לנו פונקצית משקל $w:E\to\mathbb{R}$, שאפשר לחשוב עליה כ"אורכי קשתות" (אבל במקרה הכללי נרשה גם אורכים שליליים). עבור זוג צמתים $s,t\in V$, נרצה למצוא את ה"מרחק" ביניהם. עבור כל מסלול (פשוט או לא) $s=v_0,v_1,\ldots,v_k=t$, נחשב את המשקל הכולל שלו $w(P)=\sum_{i=1}^{k}w(v_{i-1}v_i)$. נסמן "$P:s\rightsquigarrow t$" עבור "$P$ הוא מסלול מ־s ל־t", ונרצה לחשב את האינפימום $\mathrm{dist}(s,t)=\inf_{P:s\rightsquigarrow t}w(P)$.

מדוע אינפימום ולא מינימום? ראשית, יכול להיות שאין כלל מסלול בגרף מ־s ל־t, ואז מגדירים $\mathrm{dist}(s,t)=+\infty$. שנית, נבחן את המקרה שיש מסלול $s=v_0,v_1,\ldots v_k=t$ אשר מכיל מעגל שלילי: נניח שעבור i,j מתאימים מתקיים $v_i=v_j$, וכן $\sum_{l=i+1}^{j}w(v_{l-1}v_l)<0$. במקרה כזה, אפשר לעשות מסלולים יותר ויותר קלים (המסלולים יהיו עם יותר ויותר צמתים אבל עם משקל כולל קטן), ע"י כך שנחזור במסלול פעמים נוספות על המעגל $v_i,\ldots,v_j$. במקרה כזה $\mathrm{dist}(s,t)=-\infty$. בכל המקרים האחרים המשקל הקטן ביותר הוא סופי, ויש מסלול (לפחות אחד) שמשיג אותו.

טענה (השגת משקל): אם יש מסלול מ־s ל־t, ואין מסלול כזה המכיל מעגל שלילי, אז יש מסלול חסר מעגלים אשר משיג את המשקל המינימלי, שהוא סופי.

הוכחה: ראשית, מראים שבחישוב המשקל (כאינפימום) אין צורך להתחשב במסלולים בעלי מעגלים. אם $s=v_0,v_1,\ldots,v_k=t$ הוא מסלול עבורו $v_i=v_j$, אז נסיר את המעגל, ז"א נבחן את המסלול $s=v_0,\ldots,v_i,v_{j+1},\ldots,v_k=t$ זה עדיין מסלול מ־s ל־t, וסכום המשקלות עליו אינו גדול יותר מזה של המסלול המקורי. מכיוון שמספר הצמתים פוחת כל פעם שמסירים מעגל, בסוף נגיע למסלול חסר מעגלים שסכום המשקלות שלו לא־עולה על זה של המסלול המקורי.

כל שנותר הוא לשים לב שמספר המסלולים חסרי המעגלים הוא סופי (חסום ע"י $|V|!$), ולכן יש מסלול שמשיג את המשקל המינימלי (הנחנו גם שקבוצת המסלולים לא ריקה). המשקל המתקבל סופי מכיוון שהוא שווה לסכום המשקלות של מסלול ספציפי. ■

שימו לב שמהטענה הזו נובע בפרט ש־$\text{dist}(s,s)$ תמיד יהיה או 0 או $-\infty$. לפני שנמשיך, ננסח עוד טענה מסוג "מבצעים ניתוח במסלול" שתועיל לנו הרבה בהמשך.

טענה (תתי־מסלולים קלים): אם $P = \langle v_0, v_1, \dots, v_k \rangle$ מסלול קל ביותר מ־v_0 ל־v_k, אז לכל i,j המקיימים $0 \le i \le j \le k$, תת־המסלול $P_{ij} = \langle v_i, v_{i+1}, \dots, v_j \rangle$ הוא מסלול קל ביותר מ־v_i ל־v_j.

כמו כן, אם $Q = \langle v_i, u_1, \dots, u_l, v_j \rangle$ מסלול קל ביותר אחר מ־v_i ל־v_j (לא חייב להיות עם אותו מספר צמתים), אז המסלול $P' = \langle v_0, \dots, v_i, u_1, \dots, u_l, v_j, \dots, v_k \rangle$ המתקבל מהחלפת P_{ij} ב־Q הוא גם מסלול קל ביותר מ־v_0 ל־v_k.

הוכחה: נניח ש־$Q = \langle v_i, u_1, \dots, u_l, v_j \rangle$ הוא מסלול מ־v_i ל־v_j שעבורו $w(Q) < w(P_{ij})$. נגדיר את P' כמסלול המתקבל מהחלפת P_{ij} ב־Q, ז"א $P' = \langle v_0, \dots, v_i, u_1, \dots, u_l, v_j, \dots v_k \rangle$. נקבל $w(P') = w(P) - w(P_{ij}) + w(Q) < w(P)$, סתירה למינימליות של P. כמו כן, אם P_{ij} מסלול קל ביותר (כמו שהרגע הראינו) ו־Q מסלול קל ביותר אחר (ואז $w(Q) = w(P_{ij})$), אז אותו חישוב מראה שמתקיים $w(P') = w(P)$, ז"א ש־P' גם מסלול קל ביותר מ־v_0 ל־v_k. ■

כעיקרון ישנם שלושה סוגים מרכזיים של בעיות מסלול קל ביותר. אפשר לחפש עבור זוג בודד $s,t \in V$ את משקל המסלול הקל ביותר מ־s ל־t. אפשר לחפש את כל משקלי המסלולים הקלים ממקור בודד s לכל הצמתים הנגישים ממנו, בעיה מרכזית בתחום הנקראת Single Source Shortest Path. לבסוף, אפשר לנסח את בעיית ה־All Pairs Shortest Path, שבה מבקשים את כל משקלי המסלולים הקלים בין כל זוג צמתים בגרף.

אנחנו נרצה בד"כ גם למצוא את המסלולים הקלים עצמם (עבור משקלים סופיים). בגרסה של מקור בודד, אפשר לנסות לייצג את כל המסלולים ע"י עץ מושרש יחידי (עם s כשורש). המשפט הבא אומר שכאשר המשקלים סופיים אז קיים עץ כזה.

משפט (עץ מסלולים קלים) עם הוכחה כתרגיל: אם אין מעגל ממשקל שלילי נגיש מ־s, אז קיים עץ עם שורש s הכולל את כל הצמתים הנגישים מ־s, כך שכל מסלול מהשורש s לצומת v בעץ הוא מסלול קל ביותר בגרף G.

רעיון ההוכחה (השלימו כתרגיל): אפשר לבנות את העץ בצורה "אלגוריתמית". מתחילים מעץ הכולל רק את השורש s, וכל פעם שיש צומת v נגיש מ־s שאינו בעץ מבצעים את הדברים הבאים:

- יהי $s = v_0, v_1, \dots, v_k = v$ מסלול קל ביותר מ־s ל־v
- יהי i האינדקס המינימלי כך ש־v_i אינו בעץ
- מוסיפים לעץ את הצומת v_i ואת הקשת $v_{i-1}v_i$

כך ניתן להרחיב את העץ עד שיכסה את כל הצמתים הנגישים מ־s. בכל צעד כזה המסלול בעץ יהיה מסלול קל ביותר ל־v_i שנוסף לעץ, לפי הנחת האינדוקציה על v_{i-1} והטענה על תתי־מסלולים קלים. ■

באופן לא מפתיע, עץ כזה יקרא "עץ מסלולים קלים ביותר מ־s".

אלגוריתמים לבעיית המקור הבודד

כאן נראה אלגוריתמים, שעבור גרף $G = (V,E)$, פונקצית משקל $w : V \to \mathbb{R}$, ומקור $s \in V$, במידה ואין מעגל שלילי נגיש מ־s ימצאו את כל הערכים $\text{dist}(s,v)$, וכן עץ מסלולים קלים מתאים. האלגוריתמים

יתחזקו פונקציה $d:V\to\mathbb{R}\cup\{+\infty\}$ שתחסום מלמעלה את משקל המסלול הקל ביותר, שאותה "נשפר" עד שתהיה זהה למשקל המסלול הקל ביותר עצמו. עץ המסלולים הקלים ייוצג ע"י פונקציית מצביעים $p:V\to V\cup\{\mathrm{nil}\}$.

הגדרה: הפונקציה $d:V\to\mathbb{R}\cup\{+\infty\}$ תיקרא פונקציית חסם עליון, אם לכל $v\in V$ מתקיים $\mathrm{dist}(s,v)\le d(v)$. הפונקציה $p:V\to V\cup\{\mathrm{nil}\}$ תיקרא מדגימה עבור d, אם $\{uv:u=p(v)\}$ היא תת־קבוצה של קשתות הגרף אשר מתאימה לעץ עם שורש s, כל צומת v עם $d(v)<+\infty$ יהיה בעץ הנ"ל, ולכל זוג $p(v),v$ מתקיים $w(p(v),v)\le d(v)-d(p(v))$.

טענה (הדגמת משקלים): אם v צאצא של u לפי הפונקציה p (ז"א שהפעלת p על v מספיק פעמים תביא אותנו ל־u), אז ההפרש $d(v)-d(u)$ חוסם את סכום המשקלות על המסלול על העץ מ־u ל־v.

הוכחה: מבצעים אינדוקציה על המרחק (הפרש העומק) על העץ בין הצמתים המדוברים (שווה למספר הפעמים שמפעילים את p להגעה מ־v ל־u). הבסיס הוא $u=v$, שאז המדובר על מסלול חסר קשתות, ואכן $d(v)-d(u)=0$.

עבור צעד האינדוקציה, מסמנים את המסלול ב־$\langle u_0,\ldots,u_k\rangle$ כאשר $u_0=u$ ו־$u_k=v$. נזכיר שבדומה להגדרות בפרקים הראשונים בנוגע לעצי חיפוש, מתקיים $p(u_i)=u_{i-1}$ לכל $1\le i\le k$. לפי הנחת האינדוקציה $d(u_{k-1})-d(u)$ הוא לפחות סכום המשקלות $\sum_{i=1}^{k-1}w(u_{i-1}u_i)$. אם מוסיפים למסלול את הקשת $u_{k-1}u_k$ עם הנתון $w(u_{k-1}u_k)\le d(u_k)-d(u_{k-1})$, מקבלים $\sum_{i=1}^{k}w(u_{i-1}u_i)\le d(u_k)-d(u_0)$ כנדרש. ■

מהטענה על הדגמת משקלים נובע בפרט ש־$\mathrm{dist}(u,v)$ חסום ע"י $d(v)-d(u)$, בגלל שקיים לפחות מסלול אחד בין הצמתים שסכום משקלי הקשתות שלו חסום ע"י ההפרש הנ"ל. מכאן הדרך קצרה לטענה הבאה.

טענה מיידית (נכונות p): אם $d(v)=\mathrm{dist}(s,v)$ לכל $v\in V$ ו־p פונקציה מדגימה עבור d, אז p מתאימה לעץ מסלולים קלים ביותר, כי היא לא יכולה להדגים מסלולים עם משקלים קטנים מהמינימלים. ■

עתה נגדיר מושג של שיפור לפונקציית חסם עליון. נניח שנתונים לנו הפונקציה d והפונקציה המדגימה p. עבור קשת $uv\in E$ נגדיר את הפרוצדורה הבאה.

נסיון שיפור לפי קשת $uv\in E$

משתנים: גרף (בד"כ מכוון) $G=(V,E)$, פונקציית חסם עליון $d:V\to\mathbb{R}\cup\{+\infty\}$ ופונקציה מדגימה $p:V\to V\cup\{\mathrm{nil}\}$ (ביחס ל־d ולצומת מקור $s\in V$)

קלט: קשת $uv\in E$ והמשקל שלה $w(uv)\in\mathbb{R}$

- **אם** $d(v)>d(u)+w(uv)$ אז
  - מציבים $d(v)\leftarrow d(u)+w(uv)$ (שיפור d לפי uv)
  - מציבים $p(v)\leftarrow u$ (עדכון p ל־d החדש)
- **אחרת** לא עושים כלום (הקשת uv אינה משפרת את d)

הלמה הבאה מראה לנו ששום דבר לא "נהרס" כתוצאה משיפור.

למה (שימור בשיפור): אם d היא פונקצית חסם עליון, ובצענו נסיון שיפור לפי uv, אז לאחר מכן הפונקציה החדשה המתקבלת d' היא גם פונקציה חסם עליון; במקרה שבו אין מעגל שלילי נגיש מ־s ו־p היתה מדגימה עבור d, הפונקציה החדשה p' היא פונקציה מדגימה עבור d'.

הוכחה: יש מה להוכיח רק עבור המקרה ש־uv משפרת את d. לפי ההנחות יש מסלול מ־s ל־u שמשקלו (לפי סכום המשקלות) אינו עולה על $d(u)$. אם משרשרים למסלול זה את הקשת uv בסופו, אז מתקבל מסלול מ־s ל־v שמשקלו אינו עולה על $d(u)+w(uv)$, ז"א ש־$d'(v)$ עדיין חוסמת את $\mathrm{dist}(s,v)$ (ולא שינינו ערכים אחרים של d).

עבור החלק השני, נניח שההצבה $p(v) \leftarrow u$ גורמת למעגל. זה אומר שהמסלול מהשורש ל־u המודגם ע"י פנקציה p המקורית כבר כולל את v. לפי טענת ההדגמה למעלה, סכום משקלי הקשתות מ־v ל־u חסום ע"י $d(u)-d(v)$. כמו כן נתון לנו ש־$d(v)-d(u) > w(uv)$. זה אומר שסכום המשקלות על המעגל הכולל את המסלול מ־v ל־u בתוספת הקשת uv הוא בעל סכום משקלות קטן ממש מ־$(d(u)-d(v))+(d(v)-d(u))=0$, בסתירה להנחה שאין מעגלים שליליים נגישים מ־s.

שימור שאר התכונות של p (ההתאמה לעץ עם שורש s והדרישה ביחס לערכי d) אינו קשה להוכחה (בהינתן שימור חוסר המעגליות שהוכחנו הרגע), ומושאר כתרגיל. ■

אפשר עתה לנסח אלגוריתם גנרי למציאת מסלולים קלים ביותר ממקור בודד, כולל בניית עץ מתאים.

אלגוריתם גנרי עבור מסלולים קלים ממקור בודד

קלט: גרף (בד"כ מכוון) $G=(V,E)$, פונקצית משקל $w:E\to\mathbb{R}$ וצומת מקור $s\in V$

פלט: פונקצית משקל מסלולים קלים ביותר $d:V\to\mathbb{R}\cup\{+\infty\}$ ופונקצית מצביעים $p:V\to V\cup\{\mathrm{nil}\}$ המתאימה לעץ מסלולים קלים ביותר

- **אתחול:** לכל $v\in V$ מציבים $d(v)\leftarrow+\infty$ ו־$p(v)\leftarrow\mathrm{nil}$, ומציבים $d(s)\leftarrow 0$
- **כל עוד** קיימות קשתות משפרות בגרף
  - **מבצעים** שיפור לפי אחת הקשתות המשפרות

אפשר להוכיח שאם אין צמתים עם משקל מינימלי מ־s של $-\infty$ (עקב מעגלים נגישים עם סכום משקלות שלילי) אז האלגוריתם תמיד יעצור בסוף. לא נעשה את זה כאן, הדבר נובע מכך שיש מספר אפשרויות סופי ליצור סכומים של משקלות (ומראים גם שכל ערכי d במהלך האלגוריתם יהיו סכומים כאלה, תוך שימוש בשיטות ההוכחה של למת השימור בשיפור עבור p).

הדבר החשוב באמת באלגוריתם הוא ההוכחה שכאשר יש עצירה, ז"א כאשר אין קשתות משפרות, אז d היא באמת פונקצית משקל המסלולים הקלים מ־s (ואז הטענה על נכונות p נכנסת לפעולה גם). ראשית נראה שעצירה יכולה להיות רק כאשר כל הצמתים הנגישים מ־s הושגו ואין מעגלים שליליים נגישים מ־s.

למה (צמתים מושגים): כאשר האלגוריתם עוצר, לכל צומת v שיש אליו מסלול מ־s מתקיים $d(v) < +\infty$.

הוכחה: אם $d(v) = +\infty$, אז בהכרח עבור המסלול $\langle u_0, u_1 \ldots, u_k \rangle$ מ־s ל־v קיים אינדכס i שעבורו $d(u_i) = +\infty$ אבל $d(u_{i-1}) < +\infty$. בפרט הקשת $u_{i-1}u_i$ היא קשת משפרת, בסתירה. ■

למה (אין עצירה במעגל שלילי): אם האלגוריתם עוצר, אז אין מעגלים שליליים נגישים מ־s.

הוכחה: נניח ש־$u_1, \ldots, u_k$ מעגל שלילי. ז"א שכל $u_{i-1}u_i$ היא קשת וגם $u_k u_1$ קשת, ומתקיים בנוסף $w(u_k u_1) + \sum_{i=2}^{k} w(u_{i-1}u_i) < 0$. אין לנו קשתות משפרות, ולכן סכום הקשתות הנ"ל חוסם מלמעלה את סכום ההפרשים $(d(u_1) - d(u_k)) + \sum_{i=2}^{k}(d(u_i) - d(u_{i-1})) = 0$. קיבלנו סתירה. שימו לב שהשתמשנו כאן בלמה על צמתים מושגים, אחרת לא היינו יכולים לעשות את החישוב כי היו עלולים להיות בו ערכי אינסוף. ■

עכשיו אפשר להוכיח את הנכונות.

למה (נכונות כאשר אין שיפורים): כאשר האלגוריתם עוצר, d היא אכן פונקצית משקלי המסלולים הקלים ביותר מ־s.

הוכחה: תהי $U \subseteq V$ קבוצת הצמתים עבורה $d(v)$ מקבלת את $\mathrm{dist}(s, v)$. בפרט $s \in U$ (לפי הלמה הקודמת s אינו במעגל שלילי, והצבנו באתחול $d(s) = 0$). אם יש צמתים שאינם ב־U, אז ניקח מסלול קל ביותר מ־s לצומת כזה (אנחנו כבר יודעים שאין מעגלים שליליים נגישים מ־s, ולא יכולה להיות טעות עבור צומת לא נגיש כלל מ־s, ולכן יש מסלול שמשיג את המשקל). בפרט קיימת קשת uv על המסלול שעבורה $u \in U$ ו־$v \in V \setminus U$. זה אומר ש־$d(u) = \mathrm{dist}(s, u)$, אבל $d(v) > \mathrm{dist}(s, v)$. כמו כן, תת־המסלול מ־$s$ ל־u וכן תת־המסלול מ־s ל־v הם מסלולים קלים ביותר, לפי הטענה על תתי־מסלול של מסלול קל ביותר. זה אומר בפרט ש־$w(uv) = \mathrm{dist}(s, v) - \mathrm{dist}(s, u) < d(v) - d(u)$. אבל מזה נובע שהקשת הנ"ל משפרת, בסתירה. ■

עכשיו אפשר לנסח אלגוריתם ספציפי ראשון.

אלגוריתם בלמן־פורד Bellman-Ford

קלט: גרף (בד"כ מכוון) $G = (V, E)$, פונקצית משקל $w : E \to \mathbb{R}$ וצומת מקור $s \in V$

פלט: פונקצית משקל מסלולים קלים ביותר $d : V \to \mathbb{R} \cup \{+\infty\}$ ופונקצית מצביעים $p : V \to V \cup \{\mathrm{nil}\}$ המתאימה לעץ מסלולים קלים ביותר

- **אתחול:** לכל $v \in V$ מציבים $d(v) \leftarrow +\infty$ ו־$p(v) \leftarrow \mathrm{nil}$, ומציבים $d(s) \leftarrow 0$
- **מבצעים** $|V| - 1$ פעמים
  - **לכל** קשת $uv \in E$ **מבצעים** פעם אחת נסיון שיפור לפי uv
- **אם** עדיין יש קשתות משפרות אז עוצרים עם פלט "יש מעגל שלילי", **אחרת** מסיימים

זמן הריצה הוא בבירור $O(|V||E|)$. אח"כ נראה אלגוריתם יותר מהיר, אבל פחות כללי (הוא ידרוש שלא יהיו קשתות עם משקל שלילי). נראה עתה נכונות.

משפט (נכונות): אלגוריתם בלמן־פורד יפלוט שיש מעגל שלילי אם ורק אם יש מעגל כזה נגיש מ־s, ואחרת בסופו d תתן ערכים נכונים עבור המשקלים (ולכן גם p תתאים לעץ מסלולים קלים ביותר).

הוכחה: אם יש מעגל שלילי נגיש מ־s, אז לפי למת אי העצירה במעגל שלילי האלגוריתם אכן יישאר בסוף עם קשתות משפרות, ויפלוט את הפלט הנכון.

אם אין מעגל שלילי נגיש, אז זה אומר, לפי הטענה על השגת משקל, שעבור כל $v \in V$ נגיש מ־s יש מסלול $\langle v_0, v_1, \ldots, v_k \rangle$ חסר מעגלים (ולכן עם k חסום ע"י $|V| - 1$) שסכום משקלות קשתותיו שווה למשקל המינימלי. לסיום נראה באינדוקציה על i שלאחר i סבבי שיפורים $d(v_i)$ יקבל ערך נכון.

בסיס האינדוקציה הוא $i = 0$, כאשר $d(s)$ מקבל ערך נכון "0" כבר בשלב האתחול.

עבור המעבר, נשים לב שלכל $1 \le i \le k$ מתקיים $w(v_{i-1}v_i) = \mathrm{dist}(s, v_i) - \mathrm{dist}(s, v_{i-1})$ (לפי הטענה על תתי־מסלולים קלים ביותר – השתמשנו בטיעון הזה גם בהוכחת הנכונות של האלגוריתם הגנרי). אם בסוף הסבב ה־$i-1$ התקיים כבר $d(v_{i-1}) = \mathrm{dist}(s, v_{i-1})$, אז במהלך הסבב ה־$i$, כאשר ננסה לשפר לפי הקשת $v_{i-1}v_i$, הפונקציה d תקבל את הערך הנכון עבור v_i. ■

עתה נראה אלגוריתם מהיר יותר עבור המקרה ללא קשתות שליליות, האלגוריתם של דיקסטרה Dijkstra. אפשר לראות אותו כסוג של "חיפוש לרוחב מוכלל", כאשר במקום לקחת את "שכבת הקשתות הבאה", ניקח כל פעם את הקשת שתתן את סך המשקל הכי קטן מ־s לצומת החדש שהוכנס (אם כל המשקלות בגרף שווים ל־1 אז זה יוצא זהה לחיפוש לרוחב – על מנת לראות זאת, השוו את האלגוריתם למטה עם הגרסה של חיפוש לרוחב בניסוח שמתייחס לערך d מינימלי שם). אפשר גם לראות את האלגוריתם כאלגוריתם חמדני. בהוכחת הנכונות שנראה כאן, נכונות האלגוריתם הגנרי תשחק תפקיד יותר ישיר מזה ששיחקה עבור אלגוריתם בלמן־פורד.

אלגוריתם דיקסטרה Dijkstra לגרף ללא משקלים שליליים

קלט: גרף (בד"כ מכוון) $G = (V, E)$, פונקצית משקל $w : E \to \mathbb{R}^+$ וצומת מקור $s \in V$

פלט: פונקצית משקל מסלולים קלים ביותר $d : V \to \mathbb{R} \cup \{+\infty\}$ ופונקצית מצביעים $p : V \to V \cup \{\mathrm{nil}\}$ המתאימה לעץ מסלולים קלים ביותר

- **אתחול:** לכל $v \in V$ מציבים $d(v) \leftarrow +\infty$ ו־$p(v) \leftarrow \mathrm{nil}$
- מציבים $d(s) \leftarrow 0$, וכן $Q \leftarrow V$ (Q היא קבוצת הצמתים שעוד לא "התגלו", s יצא ממנה ראשון)
- **כל עוד** Q מכיל צמתים עם ערך d סופי
  - יהי $v \in Q$ צומת עם $d(v)$ מינימלי
  - **לכל** $vw \in E$ **מבצעים** נסיון שיפור לפי vw
  - מוציאים את v מ־Q

זמן הריצה תלוי במימוש Q. אם המימוש הוא בערימת מינימום, אז משלמים $O(|V|)$ בשביל האכלוס (ל־d ההתחלתית יש רק ערך אחד שונה מ־$+\infty$), ו־$O(\log(|V|))$ לכל הוצאה ולכל נסיון שיפור לפי קשת (אם הצליח לשפר). סה"כ יוצא זמן ריצה $O(|V|+|E|\log(|V|))$ (בגרף לא מכוון כל קשת משתתפת בנסיון שיפור לא יותר מפעם אחת בכל "כיוון").

הלמה הבאה מזכירה קצת את השמורה של המונוטוניות עבור גרסת התור של חיפוש לרוחב.

למה (מונוטוניות המשקלים): לעולם לא יהיה מחוץ ל־Q צומת עם ערך d גדול מזה של צומת כל שהוא בתוך Q. על כן, אם במהלך האלגוריתם מסדרים את הצמתים לפי סדר הוצאתם מ־Q, אז סדרת ערכי d המתאימה תהיה מונוטונית לא־יורדת.

הוכחה: מוכיחים את השמורה באינדוקציה על מספר הצעדים. טענת הבסיס נכונה באופן ריק (כל הצמתים נמצאים ב־Q).

עבור המעבר, שימו לב שצומת v המוצא מ־Q הוא בעל ערך d מינימלי מבין אלו ב־Q, ולכן בשלב זה עדיין לכל צמתי Q יהיו ערכי d שאינם קטנים מכל אלו של הצמתים שהוצאו מ־Q. עם זאת, לאחר ההוצאה מבצעים שיפור על הקשתות היוצאות מ־v. אולם מכיוון שהנחנו שאין קשתות עם משקלים שליליים, גם שיפורים אלו לא יכולים להביא את ערכי d של הצמתים ב־Q להיות נמוכים יותר מזה של v, או מהצמתים האחרים שאינם ב־Q (כי $d(v)$ הוא מקסימלי מביניהם לפי הנחת האינדוקציה). ■

מסקנה מיידית: מהרגע שצומת v יוצא מ־Q, הערך $d(v)$ נשאר קבוע עד סוף האלגוריתם (מכיוון שלאחריו יהיו שיפורים רק עבור קשתות מצמתים עם ערכי d שאינם קטנים מזה שלו). ■

משפט (נכונות): בסוף ההרצה של אלגוריתם דיקסטרה, ערכי d יכילו את משקלי המסלולים הקלים ביותר מ־s.

הוכחה: נראה שאין קשתות משפרות בסוף האלגוריתם, ואז נשתמש בנכונות האלגוריתם הגנרי.

לשם כך נראה באינדוקציה את קיום השמורה האומרת שבכל רגע נתון אין קשתות משפרות פרט לאלו הפנימיות ב־Q. הבסיס הוא מייד לאחר שהוצאנו את s מ־Q ועדכנו את ערכי d של השכנים שלו.

עבור המעבר, נשים לב שלכל צומת v שהוצאנו מ־Q ביצענו באותו זמן נסיון שיפור לכל הקשתות היוצאות. מכיוון שאח"כ (לפי המסקנה המיידית מקודם) הערך $d(v)$ לא קטן יותר (וערכי d של אף צומת לא יכולים לגדול), כל הקשתות היוצאות מ־v ישארו לא־משפרות.

כשהאלגוריתם מסתיים, נותרים ב־Q (אם בכלל) רק צמתים עם ערך d של $+\infty$, שאינם נגישים מ־s (אחרת חייבת להיות קשת מצומת שאינו ב־Q לצומת עם ערך $+\infty$, בסתירה לכך שאין קשתות משפרות מתוך הצמתים שהוצאו מ־Q). מכיוון שגם בין צמתים עם ערך d של $+\infty$ אין קשתות משפרות, קיבלנו שלא נותרות קשתות משפרות בגרף. ■

אלגוריתם לבעיית כל הזוגות

על מנת למצוא את משקלי המסלולים הקלים ביותר לכל זוגות הצמתים בגרף בבת אחת, שוב נשתמש בגישה של פונקציה חוסמת עליונה, רק שנגדיר אותה עבור כל הזוגות במקביל.

הגדרה: פונקציה $d: V\times V\to\mathbb{R}\cup\{+\infty\}$ תיקרא פונקציה חוסמת עליונה אם לכל $u,v\in V$ מתקיים $d(u,v)\geq \mathrm{dist}(u,v)$.

כאשר יש לנו חסם עליון על כל זוג צמתים, אז אפשר לנצל את הידע הזה על מנת לנסות לשפר לא רק באמצעות בדיקה לפי קשתות שמגיעות ל־v, אלא גם באמצעות בדיקת מסלולים אלטרנטיבים לפי צמתי אמצע. הפרוצדורה הבאה מתארת את זה באופן פורמלי.

נסיון שיפור חסם המשקל מ־u ל־v דרך צומת r

משתנים: גרף (בד"כ מכוון) $G=(V,E)$, פונקצית חסם עליון $d : V \times V \to \mathbb{R} \cup \{+\infty\}$

קלט: שלושה צמתים $u, r, v \in V$

- **אם** $d(u,v) > d(u,r) + d(r,v)$ אז
  - מציבים $d(u,v) \leftarrow d(u,r) + d(r,v)$ (שיפור $d(u,v)$ דרך r)
- **אחרת** לא עושים כלום (מעבר דרך r אינו משפר את $d(u,v)$)

לא הגדרנו כאן פונקצית מצביע שתתן לנו את המסלולים עצמם. הסיבה היא שאנחנו לא ננסה להוכיח שימור במקרה הכללי עבורה (הבעיה היא בתכונת חוסר המעגלים). בקרוב נראה אלגוריתם שמבצע את נסיונות השיפור בסדר מסויים. עבור האלגוריתם הספציפי אפשר לתחזק פונקצית מצביעים, למרות שלא נראה את זה כאן. כרגע נוכיח למת שימור, ולמת נכונות כאשר אין שיפורים ומתקיימים תנאים נוספים.

למה (שימור בשיפור): אם d היתה פונקציה חוסמת עליונה לפני נסיון שיפור, אז היא תהיה פונקציה חוסמת עליונה גם אחריו.

תקציר הוכחה: יש מה להוכיח רק כאשר מעבר דרך r באמת שיפר את $d(u,v)$. במקרה הזה ניקח מסלול P' מ־u ל־r אשר משקלו חסום ע"י $d(u,r)$ (יש כזה לפי ההנחה ש־d חוסם את משקל המסלול הקל ביותר האמיתי), ומסלול P'' מ־r ל־v אשר משקלו חסום ע"י $d(r,v)$. השרשור $P'P''$ נותן מסלול מ־u ל־v שמשקלו חסום ע"י $d(u,r)+d(r,v)$, ולכן משקל המסלול הקל ביותר מ־u ל־v חסום ע"י סכום זה. ■

למה (נכונות כשאין שיפורים): אם d היא פונקציה חוסמת עליונה שאין עבורה שיפורים, ובנוסף מקיימת $d(v,v)=0$ לכל $v \in V$ ו־$d(u,v) \leq w(uv)$ לכל קשת $uv \in E$, אז $d(u,v) = \mathrm{dist}(u,v)$ לכל $u,v \in V$. בפרט, אין במקרה כזה מעגלים שליליים בגרף.

הוכחה: נקבע צומת כל שהוא $s \in V$ ונגדיר $d_s : V \to \mathbb{R} \cup \{+\infty\}$ לפי $d_s(v) = d(s,v)$. זאת תהיה פונקציה חוסמת עליונה לפי ההגדרה של תת־הפרק הקודם עבור משקלי המסלולים הקלים מ־s לאיברי V. מכך של־d לא היו שיפורים, בתוספת הנתון על $d(u,v)$ עבור קשתות $uv \in E$, נובע של־d_s אין שיפורים לפי קשתות כפונקציה חוסמת עליונה של משקלים ממקור בודד: אם $uv \in E$, אז $d_s(v) = d(s,v) \leq d(s,u) + d(u,v) \leq d(s,u) + w(uv) = d_s(u) + w(uv)$. נוסיף לכך את הנתון $d_s(s)=0$, ועכשיו מתקיימים לנו כל התנאים הדרושים לטענת הנכונות כשאין שיפורים עבור האלגוריתם הגנרי למקור בודד (כולל למות המובילות אליה, כגון למת אי העצירה במעגל שלילי). מכאן שלכל v מתקיים $d(s,v) = d_s(v) = \mathrm{dist}(s,v)$. מכיוון שאנחנו יכולים לטעון את זה לכל $s \in V$, קיבלנו את נכונות d עבור כל זוגות הצמתים. ■

עתה נציג את אלגוריתם פלויד־וורשל למציאת כל משקלי המסלולים הקלים ביותר בגרף. הוא יהיה דוגמה ראשונה לטכניקה אלגוריתמית שנלמד עליה בהמשך, הקרויה תכנות דינמי.

אלגוריתם פלויד־וורשל Floyd-Warshall למשקלי מסלולים קלים ביותר

קלט: גרף (בד"כ מכוון) $G = (V, E)$, פונקצית משקל $w : E \to \mathbb{R}$

פלט: פונקצית משקל מסלולים קלים ביותר $d : V \times V \to \mathbb{R} \cup \{+\infty\}$

- **אתחול: לכל** זוג צמתים $u, v \in V$
  - **אם** $u = v$ אז $d(u, v) \leftarrow 0$
  - **אחרת, אם** $uv \in E$ אז $d(u, v) \leftarrow w(uv)$
  - **אחרת** $d(u, v) \leftarrow +\infty$
- **לכל** $r \in V$ (בסדר שרירותי קבוע) מבצעים
  - **לכל** זוג צמתים $u, v \in V$ **מבצעים** נסיון שיפור של $d(u, v)$ דרך r
- **אם** יש v עבורו $d(v, v) < 0$ **אז** עוצרים עם פלט "יש מעגל שלילי", **אחרת** מסיימים

עבור חישוב זמן הריצה, נשים לב שנסיון שיפור בודד לוקח $O(1)$, ולכן זמן הריצה הכולל הוא $O(|V|^3)$.

להוכחת הנכונות, נסמן $V = \{v_1, \ldots, v_n\}$, ונניח שהלולאה על $r \in V$ רצה לפי הסדר הנ"ל (מתחילה ב־$r = v_1$ וגומרת ב־$r = v_n$). עתה נגדיר את $\mathrm{dist}_i(u, v)$ בתור האינפימום של משקלי המסלולים מ־u ל־v, כאשר מתחשבים רק במסלולים שכל צמתי האמצע שלהם (לא כולל u, v עצמם) הם מתוך הקבוצה $\{v_1, \ldots, v_i\}$.

לשם הדגמה, שימו לב מהו $\mathrm{dist}_0(u, v)$. כאן אנחנו מגבילים את עצמנו למסלולים מאורך 0 (אשר נותנים $\mathrm{dist}_0(u, v) = 0$ עבור $u = v$), ומסלולים בעלי קשת בודדת (אשר נותנים $\mathrm{dist}_0(u, v) = w(uv)$ כאשר $uv \in E$). במקרים שלא צוינו כאן $\mathrm{dist}_0(u, v) = +\infty$. זהו בדיוק האתחול של $d(u, v)$ באלגוריתם. כמו כן שימו לב שמתקיים $\mathrm{dist}_n(u, v) = \mathrm{dist}(u, v)$ מיידית מההגדרות.

נגדיר את $\mathrm{dist}'_i(u, v)$ בדומה ל־$\mathrm{dist}_i(u, v)$, עם ההבדל שמתחשבים רק במסלולים פשוטים (ללא חזרות/מעגלים) בעת חישוב האינפימום.

טענה מיידית (פשטות מסלולים): כאשר אין מעגלים שליליים, אז בדיוק כמו בהוכחת הטענה בתחילת הפרק על השגת משקל, יתקיים $\mathrm{dist}_i(u, v) = \mathrm{dist}'_i(u, v)$ לכל i, u ו־v. ■

למה (נכונות דינמית): לאחר האיטרציה על v_i באלגוריתם פלויד־וורשל, לכל $u, v \in V$ יתקיים אי־השוויון $d(u, v) \le \mathrm{dist}'_i(u, v)$.

הוכחה: באינדוקציה על i. הבסיס הוא מה שציינו למעלה, ש־$d(u, v)$ מאותחל להיות זהה ל־$\mathrm{dist}_0(u, v)$ לכל $u, v \in V$.

עבור צעד המעבר מ־$i-1$ ל־i, צריך להוכיח משהו רק אם $\mathrm{dist}'_i(u,v) < +\infty$. נשים לב שמסלול פשוט מ־$u$ ל־v לא יכול להכיל את v_i יותר מפעם אחת. אם המסלול המשיג את $\mathrm{dist}'_i(u,v)$ לא עובר דרך v_i אז $\mathrm{dist}'_i(u,v) = \mathrm{dist}'_{i-1}(u,v)$, ואז הנחת האינדוקציה נותנת לנו את המבוקש מיידית.

אחרת, נסמן ב־P את המסלול דרך v_i אשר משיג את $\mathrm{dist}'_i(u,v)$. נציג אותו כשרשור של שני מסלולים, $P = P'P''$, כאשר P' תת־המסלול מ־u ל־v_i ו־P'' תת־המסלול מ־v_i ל־v. גם P' וגם P'' מקיימים שכל צמתיהם הפנימיים נמצאים ב־$\{v_1, \ldots, v_{i-1}\}$, ולכן לפי הנחת האינדוקציה בסוף האיטרציה ה־$i-1$ התקיים $d(u,v_i) \le w(P')$ ו־$d(v_i,v) \le w(P'')$. על כן, כשעשינו את נסיון השיפור מ־u ל־v דרך v_i, בהכרח הגענו למצב שבו $d(u,v) \le w(P') + w(P'') = \mathrm{dist}'_i(u,v)$. ■

משפט (נכונות): בסוף ההרצה של אלגוריתם פלויד־וורשל האלגוריתם יפלוט שיש מעגל שלילי אם הגרף מכיל מעגל כזה, ואחרת d תתן את משקלי המסלולים הקלים ביותר בין כל זוג צמתים.

הוכחה: אם אין מעגל שלילי, אז הנכונות נובעת מלמת הנכונות הדינמית עבור $i = n$, בתוספת הטענה המיידית על פשטות המסלולים.

אם יש מעגל שלילי אז נשים לב שיש מעגל פשוט כזה (ההוכחה כתרגיל, ודומה להוכחות של טענות דומות שעשינו), ונניח ש־v_i הוא הצומת בעל האינדכס הכי גבוה במעגל פשוט שלילי מסויים. יהי v_j צומת אחר באותו מעגל, בעל האינדכס השני הכי גבוה. כאשר באיטרציה ה־j עושים את ניסוי השיפור מ־v_i לעצמו דרך v_j, הערך $d(v_i,v_i)$ יהפוך להיות שלילי אם לא היה כזה קודם (זה נובע מהנכונות הדינמית עבור $\mathrm{dist}'_{j-1}(v_i,v_j)$ ו־$\mathrm{dist}'_{j-1}(v_j,v_i)$, כי המעגל ניתן להצגה כשרשור של שני המסלולים הפשוטים המתאימים). ■

תכנות דינמי

הרעיון בתכנות דינמי הוא בניית אלגוריתם הפותר בעיה בשלבים: מגדירים בנוסף לבעיה המקורית סדרה של "בעיות משנה" שמובילות אליה, ופותרים אותן בסדר כזה שבכל שלב נוכל לפתור את בעית המשנה הבאה בקלות יחסית בהסתמך על בעיות המשנה שפתרנו עד עכשיו.

אלגוריתם פלויד־וורשל הוא דוגמה לאלגוריתם כזה: אנחנו רצינו לפתור את $|V|^2$ הבעיות מהצורה של "מצא את $\mathrm{dist}(u,v)$", ולשם כך לכל $1 \le i \le |V|$ הגדרנו $|V|^2$ בעיות משנה של חסימת $\mathrm{dist}_i(u,v)$ (המשקל המינימלי המושג ע"י מסלולים דרך $\{v_1, \ldots, v_i\}$). אפשר לכתוב את האלגוריתם ככזה שממלא "טבלה" שהאינדכסים שלה הם $V \times V \times \{1, \ldots, |V|\}$, כאשר בכל פעם ממלאים את ה"שכבה" $V \times V \times \{i\}$ בהסתמך על השכבה הקודמת (זהו מקור השם "נכונות דינמית" של הלמה שהוכחה שם).

גם חלק מהאלגוריתמים החמדניים יכולים להיכתב כאלגוריתמי תכנות דינמי, כאשר עבורם חישוב בעיית המשנה הנוכחית לפי בעיות המשנה שחושבו קודם מסתמך על עיקרון חמדני פשוט.

דוגמה מהירה נוספת

הדוגמה הבאה שנראה היא בעלת מבנה "חד־מימדי" של בעיות משנה. ליתר דיוק, זו תהיה הגרסה הממושקלת של בעיית השיבוץ של קבוצת קטעים זרים מקסימלית, אשר נידונה בפרק על אלגוריתמים חמדניים. הפעם הצעד הדינמי לא יוכל להסתמך אך ורק על עיקרון חמדני פשוט.

כמו בהגדרה המקורית, יש לנו קבוצה $S = \{a_1, \ldots a_n\}$ של קטעים המאופיינים ע"י זמני התחלה וסיום $a_i = (s_i, f_i)$. כאן אבל תהיה לנו גם פונקצית משקל $w : S \to \mathbb{R}$, ונרצה למצוא תת־קבוצה $S' \subseteq S$

של קטעים זרים שנותנת מקסימום עבור $w(S') = \sum_{a\in S'} w(a)$. אפשר להניח ש־$w$ מקבלת רק ערכים חיוביים, כי קטעים עם משקלים שאינם חיוביים פשוט לעולם לא יוכנסו ל־S'.

בדומה לאלגוריתם החמדן, ראשית נמיין את הקבוצה S לפי זמני הסיום של הקטעים (בזמן ריצה $O(n\log(n))$) כך שיתקיים $f_1 \le f_2 \le \cdots \le f_n$, ולכל i נסמן ב־S_i את תת־הקבוצה $\{a_1,\ldots,a_i\}$. עתה נרצה למצוא תת־קבוצה $S'_i \subseteq S_i$ של קטעים זרים שעבורה מקבלים ערך מקסימלי ל־$w(S'_i) = \sum_{a\in S'_i} a$. לאחר פתרון כל בעיות המשנה, נחזיר את $S' = S'_n$ ואת $w(S'_n)$ בהתאמה.

עבור האלגוריתם הדינמי, יהיה נוח לחשב מראש לכל $1 \le i \le n$ ערך "קטע זר קודם אחרון" של $p_i = \max\{j : f_j \le s_i\}$ (גם כאן מניחים קטעים פתוחים ובפרט $s_i < f_i$). אם הקבוצה בביטוי ריקה אז נגדיר $p_i = 0$, כי זה נמוך מכל האינדכסים. זה מפתה לנסות אלגוריתם "דמוי מיזוג" לחישוב מהיר של כל ה־p_i, אבל ה־s_i אינם בהכרח ממויינים. על כן פשוט נשתמש בחיפוש בינארי לחישוב כל p_i לחוד, ונשקיע זמן חישוב כולל של $O(n\log(n))$.

תמצית האלגוריתם הדינמי מתמצאת בלמה הבאה.

למה (נוסחה דינמית): לכל $1 \le i \le n$ מתקיים $w(S'_i) = \max\{w(S'_{i-1}), w(S'_{p(i)}) + w(a_i)\}$. יתרה מזאת, אם $w(S'_{i-1})$ הוא איבר מקסימלי בביטוי אז אפשר לקבוע $S'_i = S'_{i-1}$, ואחרת צריך לקבוע $S'_i = S'_{p(i)} \cup \{a_i\}$.

הוכחה: אם יש תת־קבוצה של S_i (של קטעים זרים) עם משקל מקסימלי שאינה מכילה את a_i עצמו, אז זו גם תת־קבוצה של S_{i-1} שמקבלת משקל מקסימלי, ועל כן אפשר לקבוע $S'_i = S'_{i-1}$.

אחרת, $a_i \in S'_i$, ופרט ל־a_i יש להוסיף תת־קבוצה מקסימלית של קטעים שזרים ל־a_i. מכיוון ש־$f_1,\ldots,f_n$ ממויינים, הקבוצה $S_{p(i)}$ היא בדיוק תת־הקבוצה של כל הקטעים הזרים ל־a_i מתוך S_i, ולכן צריך להוסיף ל־a_i עצמו את תת־הקבוצה של $S_{p(i)}$ בעלת המשקל המקסימלי, ז"א את $S'_{p(i)}$. ■

עכשיו אפשר להציג את האלגוריתם הדינמי המלא. אנחנו נתחזק באלגוריתם את משקלי הקבוצות המקסימלים $w_i = w(S'_i)$ במקביל ל־S'_i עצמם.

אלגוריתם דינמי לחישוב קבוצת קטעים זרים מקסימלית לפי משקל

קלט: קבוצת קטעים S ופונקצית משקל $w : S \to \mathbb{R}^+$

פלט: קבוצה $S' \subseteq S$ במשקל מקסימלי שמכילה קטעים זרים זה לזה, והמשקל שלה $w' = w(S')$

- **אתחול:** ממיינים את S לפי זמני סיום, מסמנים את התוצאה $(a_1,f_1),\ldots,(a_n,f_n)$, מחשבים את $p_1,\ldots,p_n$ לפי $p_i = \max\{j : f_j \le s_i\}$, ומציבים $S'_0 \leftarrow \emptyset$ ו־$w_0 \leftarrow 0$
- **עבור** i מ־1 עד n **לפי הסדר**
  - **אם** $w_{i-1} \ge w_{p_i} + w(a_i)$
    - מציבים $S'_i \leftarrow S'_{i-1}$ ו־$w_i \leftarrow w_{i-1}$
  - **אחרת**
    - מציבים $S'_i \leftarrow S'_{p_i} \cup \{a_i\}$ ו־$w_i \leftarrow w_{p_i} + w(a_i)$

• מציבים עבור הפלט $S' = S'_n$ ו־$w' = w_n$

זמן הריצה של האלגוריתם הוא $O(n\log(n))$ עבור פעולות האתחול, ואח"כ $O(1)$ עבור כל אחת מ־n האיטרציות לפי i בלולאה (את ה־S'_i אפשר לייצג ע"י שימוש במבנה עץ, כאשר נשמור עבור S'_i את הנתון האם היא מכילה את a_i, יחד עם מצביע או לקבוצה S'_{i-1} או לקבוצה $S'_{p(i)}$). סה"כ זמן הריצה הוא $O(n\log(n))$.

טענה מיידית (נכונות): בכל שלב S'_i אכן מקבלת קבוצת קטעים זרים בעלת משקל מקסימלי, ו־w_i מקבל את $w(S'_i)$. ההוכחה היא באינדוקציה תוך כדי שימוש בלמת הנוסחה הדינמית. על כן האלגוריתם בסוף פולט תת־קבוצה של קטעים זרים ב־S בעלת משקל מקסימלי. ■

אופטימיזציה של כפל מטריצות

כאן נראה דוגמה (מאוד פשטנית) למה שבד"כ קורה "מאחורי הקלעים" במימושים של שפות תכנות עיליות. לצורך העניין נניח שיש לנו צורך למצוא מכפלה של n מטריצות $\prod_{i=1}^{n} A_i = A_1A_2\cdots A_n$ (בשפת Matlab יכול לקרות כזה דבר). נניח שעבור כפל בודד AB, כאשר A היא מגודל $p\times q$ ו־B היא מגודל $q\times r$, אנחנו משתמשים באלגוריתם הנובע מההגדרה של כפל מטריצות, המכיל pqr פעולות כפל וסדר גודל דומה של פעולות חיבור. לצורך חישובי הזמן נתעלם מפעולות החיבור, שהן זולות בהרבה מפעולות כפל (נסו להיזכר בכפל וחיבור ארוך שלמדתם בבית הספר).

עבור המכפלה $\prod_{i=1}^{n} A_i$ נניח שכל מטריצה A_i היא ממימדים $p_{i-1}\times p_i$ (ובפרט פעולת הכפל תהיה חוקית). כל פעם אנחנו יכולים לכפול רק שתי מטריצות, ולכן נשמור תוצאות ביניים להמשך הכפל. אנחנו נרצה להשתמש בחוקי האסוציאטיביות לטובתנו, ולמצוא תכנון של סדרת ההכפלות שייתן מספר מינימלי של פעולות כפל מספרים בסה"כ הכולל. ננסח כאן "אלגוריתם מתכנן" מתאים אשר נועד להרצה לפני שמתחילים לבצע את פעולות כפל המטריצות עצמן.

למשל, נניח שרוצים לכפול את ABC כאשר A היא 1×2, B היא 2×3, ו־C היא 3×4. אם כופלים קודם את AB, ז"א מבצעים "$(AB)C$", אז סך פעולות הכפל הכולל הוא $18 = (1\times 3\times 4) + (1\times 2\times 3)$. אם כופלים קודם את BC, ז"א מבצעים "$A(BC)$", אז סך פעולות הכפל הוא $32 = (2\times 3\times 4) + (1\times 2\times 4)$, הבדל ניכר. אפשר גם למצוא פעולות עם הבדלים גדולים יותר. נסו לכתוב דוגמה לשלוש מטריצות עם מספר פעולות כפל של $O(r)$ מול $O(r^2)$ עבור פרמטר נתון r.

באופן פורמלי נרצה לבנות עץ ביטוי: זהו עץ בינארי שלם (כל הצמתים הפנימיים יהיו בעלי שני בנים) מושרש ומסודר (לכל צומת פנימי יש בן ימני ובן שמאלי), שהעלים שלו מתאימים ל־n המטריצות שלנו והצמתים הפנימיים שלו מתאימים לפעולות כפל. למשל, עבור "$(AB)C$" לשורש יהיה בן שמאלי "AB" ששני בניו הם העלה "A" והעלה "B", בעוד שהבן הימני של השורש הוא העלה "C".

בעיות הביניים שנגדיר יהיו אלו של הדרך הטובה ביותר לחשב את הכפל החלקי $\prod_{t=i}^{j} A_t$ עבור כל זוג אינדכסים $1\le i\le j\le n$ (עבור $i=j$ פעולת ה"כפל" היא פשוט להחזיר את A_i, ללא שימוש בפעולות כפל מספריות). המבנה שלנו יהיה "דו־מימדי", כי הוא תלוי בשני אינדכסים.

נגדיר אם כן את $m_{i,j}$ להיות מספר פעולות הכפל המספריות האופטימלי עבור חישוב $\prod_{t=i}^{j} A_t$, ו־$T_{i,j}$ יהיה עץ ביטוי מתאים עם עלים $A_i,\ldots,A_j$. בפרט $m_{i,i}=0$ ו־$T_{i,i}$ הוא עץ בעל צומת בודד A_i. אם נסתכל על $T_{i,j}$ וניקח $i\le k<j$, אז אופציה אחת לבניית עץ עבור $T_{i,j}$ תתאים לכפל $(\prod_{t=i}^{k} A_t)(\prod_{t=k+1}^{j} A_t)$,

הווה אומר העץ שנוצר כאשר לוקחים צומת שורש ומחברים לבנים שלו (כתתי־עצים) את $T_{i,k}$ ואת $T_{k+1,j}$ בהתאמה. מכיוון שזו אחת האפשרויות, בפרט מתקיים $m_{i,j} \leq m_{i,k} + m_{k+1,j} + p_{i-1}p_kp_j$, לפי החישוב של מספר הפעולות עבור העץ המתואר כאן. מכאן אנו מקבלים את הטענה הבאה.

טענה (צעד דינמי): בהינתן הערכים $m_{i,k}$ ו־$m_{k+1,j}$ והעצים $T_{i,k}$ ו־$T_{k+1,j}$ לכל $i \leq k < j$, ניתן לחשב את $m_{i,j}$ לפי הנוסחא $\min_{i \leq k < j}\{m_{i,k} + m_{k+1,j} + p_{i-1}p_kp_j\}$. יתרה מזו, עבור בניית $T_{i,j}$, יש לקחת k שמשיג את המינימום בביטוי עבור $m_{i,j}$, ולבנות עץ שהבנים של השורש שלו הם השורשים של תת־עץ זהה ל־$T_{i,k}$ ושל תת־עץ זהה ל־$T_{k+1,j}$ בהתאמה.

הוכחה: הפעולה האחרונה בחישוב כל שהוא של $\prod_{t=i}^{j} A_t$ היא תמיד כפל של שתי מטריצות שחושבו קודם. יתרה מזו, הפירוק יהיה תמיד מהצורה $(\prod_{t=i}^{k} A_t)(\prod_{t=k+1}^{j} A_t)$ עבור k מתאים (אנחנו יכולים להשתמש כזכור באסוציאטיביות בלבד, ואין אפשרות לשנות את סדר המטריצות, או להוריד או להוסיף מהן). כמו כן, תמיד נוכל רק להקטין את מספר פעולות הכפל אם נחשב את $\prod_{t=i}^{k} A_t$ ואת $\prod_{t=k+1}^{j} A_t$ בשיטה האופטימלית האפשרית עבורם, ז"א נקבל $m_{i,j} = m_{i,k} + m_{k+1,j} + p_{i-1}p_kp_j$ עבור ה־k המתאים. מכיוון שאנחנו מעונינים במינימום פעולות, ה־k המתאים חייב להיות כזה שישיג את המינימום עבור הביטוי למעלה, ז"א ש־$m_{i,j}$ ו־$T_{i,j}$ יהיו לפי הטענה. ■

עתה ננסח את האלגוריתם הדינמי. הוא יעבוד לפי סדר עולה של $i - j$.

אלגוריתם דינמי לחישוב עץ־ביטוי אופטימלי עבור כפל מטריצות

קלט: סדרה $p_0, \ldots, p_n$ אשר מתארת את המכפלה $\prod_{i=1} A_i$, כשכל A_i היא מטריצה מגודל $p_{i-1} \times p_i$

פלט: מספר פעולות כפל קטן ביותר m עבור חישוב המכפלה הנ"ל, ועץ ביטוי מתאים T

- **אתחול: לכל** $1 \leq i \leq n$ מציבים $T_{i,i} \leftarrow (\{A_i\}, \emptyset)$ (עץ עם צומת בודד) ו־$m_{i,i} \leftarrow 0$
- **עבור** d מ־1 עד $n - 1$ **לפי הסדר**
  - **לכל** $1 \leq i \leq n - d$ מחשבים את $m_{i,i+d}$ ואת $T_{i,i+d}$ לפי טענת הצעד הדינמי
- מציבים עבור הפלט $m = m_{1,n}$ ו־$T = T_{1,n}$

מבחינת זמן ריצה, אם מממשים את מבני העצים באמצעות מבנה נתונים מקושר מתאים, אז מקבלים זמן $O(n)$ עבור האתחול, שלאחריו לכל d יהיו $n - d$ הפעולות של הצעד הדינמי. טענת הצעד הדינמי משתמשת בעצמה במינימום של d ביטויים שכל אחד מהם ניתן לחישוב בזמן $O(1)$. לאחר החישוב, בניית $T_{i,j}$ עצמו גם תהיה בזמן $O(1)$ עבור מבנה נתונים מקושר מתאים. סה"כ קיבלנו לאחר האיתחול זמן של $O(\sum_{d=1}^{n-1} d(n-d)) = O(n^3)$, הווה אומר שהאלגוריתם שלנו רץ בזמן $O(n^3)$. נשאר רק להוכיח את הנכונות.

טענה (נכונות): האלגוריתם יחזיר בסוף הריצה עץ־ביטוי אופטימלי עבור $\prod_{t=1}^{n} A_t$.

הוכחה: ההוכחה היא באינדוקציה על d שכל הערכים $m_{i,i+d}$ ו־$T_{i,i+d}$ יתאימו לחישובים האופטימלים. הבסיס $d = 0$ נובע מצעד האתחול של האלגוריתם, שנעשה לפי הדיון למעלה (לפני טענה הצעד הדינמי).

צעד המעבר הוא לפי טענת הצעד הדינמי. צריך רק להוכיח שכאשר נגיע לחשב את $m_{i,i+d}$ ואת $T_{i,i+d}$, יהיו כבר בידינו כל הערכים הדרושים $m_{i,k}$ ו־$m_{k+1,i+d}$ והעצים $T_{i,k}$ ו־$T_{k+1,i+d}$ עבור $i \leq k < i+d$. דבר זה נובע מכך שלכל ערכי k הרלוונטים מתקיים $k-i<d$ ו־$(i+d)-(k+1)<d$, ז"א שערכים אלו חושבו בצעדים עבור ערכי d קטנים מה־d מהנוכחי. ■

התאמת מחרוזות

התאמת מחרוזות sequence alignment היא גרסה מוכללת של "מרחק עריכה". הרעיון של מרחק עריכה הוא זה: נניח שיש מחרוזת מקור $A=a_1\cdots a_m \in \Sigma^*$ ומחרוזת יעד $B=b_1\cdots b_n \in \Sigma^*$ (שימו לב שלא חובה ששתי המחרוזות יהיו מאותו אורך). מהו המספר המינימלי של פעולות מחיקת אות, הכנסת אות, והחלפת אות שתביא אותנו מ־A ל־B?

באופן יותר כללי, תהיה לנו פונקצית עלות $w:(\Sigma\cup\{\epsilon\})\times(\Sigma\cup\{\epsilon\})\to\mathbb{R}$ (משמעות הסימון הוא פונקציה "דו־מקומית"; המשתנים שלה יכולים לקבל אותיות מ־Σ או את הערך המיוחד ϵ). המשמעות היא שעבור $a,b\in\Sigma$ הערך $w(a,b)$ הוא עלות ההחלפה של a ב־b, הערך $w(a,\epsilon)$ הוא עלות המחיקה של a, והערך $w(\epsilon,b)$ הוא עלות ההכנסה של אות b. אין משמעות ל־$w(\epsilon,\epsilon)$.

אם מדברים ספציפית על מרחק עריכה, הנחות סבירות הן שפונקצית העלות היא אי־שלילית, מקיימת $w(x,x)=0$ (אין עלות בהשארת אות במקום) ו־$w(x,y)+w(y,z)\geq w(x,z)$ (למשל שהחלפה ישירה של אות אינה עולה יותר מהחלפה דרך אות שלישית – במודל שלנו נתעלם מאפשרות של יותר מ"עריכה אחת" לכל אות). עבור בעיית ההתאמה שמיד נגדיר נראה אלגוריתם שעובד גם ללא הנחות אלו, אבל הן מקשרות את המושג המוגדר למושג האינטואיטיבי של מרחק עריכה.

אנחנו נגדיר הרחבה של המילה A כמחרוזת $A'\in(\Sigma\cup\{\epsilon\})^*$, כאשר אם נמחק מ־$A'$ את כל המופעים של ϵ נקבל בדיוק את A. התאמה בין A ל־B מוגדרת ע"י הרחבה $A'=a'_1\cdots a'_l$ של A והרחבה $B'=b'_1\cdots b'_l$ של B, שתיהן מאותו אורך, כך שאין i עבורו $a'_i=b'_i=\epsilon$ (בפרט יש לנו את החסמים $\max\{n,m\}\leq l\leq n+m$). העלות של ההתאמה מוגדרת ע"י $w(A',B')=\sum_{i=1}^{l}w(a'_i,b'_i)$.

המרחק $\operatorname{dist}(A,B)$ (בגרסה המוכללת) בין A ל־B הוא העלות המינימלית $w(A',B')$ של התאמה בין A ל־B (אם מתקיימות "ההנחות הסבירות" למעלה, זו באמת תהיה העלות המינימלית האפשרית לעריכת B מתוך A). נראה עתה כיצד לחשב אותו. אפשר להרחיב את הטיעונים שיובאו כאן גם למציאת ההתאמה המתאימה, וזה מושאר לכם כתרגיל.

גם כאן נגדיר מבנה "דו־מימדי" של בעיות משנה. נגדיר את הרישות של המילים $A_i=a_1\cdots a_i$ ו־$B_j=b_1\cdots b_j$. בעיות המשנה שלנו יהיו מציאת $\operatorname{dist}(A_i,B_j)$ לכל $1\leq i\leq m$ ו־$1\leq j\leq n$.

עבור בעיות המשנה נשתמש בלמה הבאה.

למה (חישוב דינמי): עבור $1<i\leq m$ ו־$1<j\leq n$ המרחק $\operatorname{dist}(A_i,B_j)$ שווה בדיוק למינימום הבא:
$$\min\{\operatorname{dist}(A_{i-1},B_{j-1})+w(a_i,b_j),\ \operatorname{dist}(A_i,B_{j-1})+w(\epsilon,b_j),\ \operatorname{dist}(A_{i-1},B_j)+w(a_i,\epsilon)\}$$

הוכחה: לכל i,j נסמן את ההתאמה המשיגה את $\operatorname{dist}(A_i,B_j)$ ב־$A'_{i,j},B'_{i,j}$ (שתי ההרחבות יכולות להיות תלויות גם ב־i וגם ב־j). הביטוי למעלה נותן את שלושת האפשרויות עבור האות האחרונה a' ב־$A'_{i,j}$ והאות האחרונה b' ב־$B'_{i,j}$. אם $(a',b')=(a_i,b_j)$ אז העלות היא $\operatorname{dist}(A_{i-1},B_{j-1})+w(a_i,b_j)$. הסיבה לכך היא שהעלות של התאמה מהצורה $(\tilde{A}a_i,\tilde{B}b_j)$ בין A_i ל־B_j היא $w(\tilde{A},\tilde{B})+w(a_i,b_j)$, ולכן המינימום יושג עבור $\tilde{A}$ ו־$\tilde{B}$ המשיגים את $\operatorname{dist}(A_{i-1},B_{j-1})$, ז"א $(\tilde{A},\tilde{B})=(A'_{i-1,j-1},B'_{i-1,j-1})$.

באופן דומה לדיון למעלה, אם $(a',b')=(\epsilon,b_j)$ אז העלות היא $\mathrm{dist}(A_i,B_{j-1})+w(\epsilon,b_j)$, ואם $(a',b')=(a_i,\epsilon)$ אז העלות היא $\mathrm{dist}(A_{i-1},B_j)+w(a_i,\epsilon)$. ■

שימו לב שהוכחת הלמה למעלה גם נותנת שיטה לחישוב ההתאמה $(A'_{i,j},B'_{i,j})$ בהסתמך על ההתאמות $(A'_{i-1,j-1},B'_{i-1,j-1})$, $(A'_{i,j-1},B'_{i,j-1})$, ו־$(A'_{i-1,j},B'_{i-1,j})$.

עתה ננסח את האלגוריתם הדינמי למציאת $\mathrm{dist}(A,B)$. מקרי הבסיס הם $\mathrm{dist}(A_i,B_0)$ ו־$\mathrm{dist}(A_0,B_j)$, ועבורם ההתאמה היחידה עם המילה הריקה ניתנת ע"י סדרת ההכנסות או ההוצאות המתאימה.

אלגוריתם דינמי לחישוב עלות התאמה (מרחק עריכה)

קלט: שתי מחרוזות $A=a_1,\ldots,a_m$ ו־$B=b_1,\ldots,b_n$ מעל אלפבית Σ, ופונקציה של עלות עריכה $w:(\Sigma\cup\{\epsilon\})\times(\Sigma\cup\{\epsilon\})\to\mathbb{R}$

פלט: עלות מינימלית d של התאמה מ־A ל־B.

- **אתחול:** מציבים $d_{0,0}\leftarrow 0$, ובנוסף:
  - **עבור** i מ־1 עד m **לפי הסדר** מציבים $d_{i,0}\leftarrow d_{i-1,0}+w(a_i,\epsilon)$ (מתקבל $d_{i,0}=\sum_{k=1}^{i}w(a_i,\epsilon)$)
  - **עבור** j מ־1 עד n **לפי הסדר** מציבים $d_{0,j}\leftarrow d_{0,j-1}+w(\epsilon,b_j)$ (מתקבל $d_{0,j}=\sum_{k=1}^{j}w(\epsilon,b_j)$)
- **עבור** i מ־1 עד m **לפי הסדר**
  - **עבור** j מ־1 עד n **לפי הסדר**
    - מציבים $d_{i,j}\leftarrow\min\{d_{i-1,j-1}+w(a_i,b_j),\ d_{i,j-1}+w(\epsilon,b_j),\ d_{i-1,j}+w(a_i,\epsilon)\}$
- מציבים עבור הפלט $d=d_{m,n}$

זמן הריצה הוא $O(mn)$, זמן הביצוע של שתי הלולאות המקוננות המרכזיות (זמן החישוב עד שמגיעים ללולאות המקוננות, $O(m+n)$, נבלע בזה). את הנכונות של האלגוריתם ניתן להוכיח מלמת החישוב הדינמי באינדוקציה – שימו לב שסדר הלולאות הוא כזה שאף ערך לא מחושב לפני שחושבו שלושת הערכים שהוא תלוי בהם. השלימו את הפרטים המלאים כתרגיל.

רשתות זרימה

רשתות זרימה הן משפחה של בעיות אופטימיזציה. לאחר שנלמד עליהן, נראה שבעיות רבות אחרות, כגון מציאת זיווג מושלם בגרף דו־צדדי, ניתנות לתיאור כמקרים פרטיים של רשתות זרימה.

הגדרות

הערה חשובה על ההגדרות: בספרות נהוגות שתי גישות להגדרת רשתות זרימה. הגישה ה"מתמטית" (בד"כ נהוגה בספרים יותר ישנים בתורת האלגוריתמים, וגם בספרים עם אוריאנטציה מתמטית) מזהה בין זרימה חיובית מ־u ל־v לבין זרימה שלילית באותה עוצמה מ־v ל־u. היא יותר קשה להבנה, אבל הנוסחאות הקשורות בחיבורי זרימות פשוטות מאוד. בגישה ה"אינטואיבית" (שתמצאו אותה בד"כ במהדורות העדכניות של הספרות בנושא) אין זרימות שליליות. בגישה זו כתיבה מדוייקת של ההוכחות מחייבת חלוקה להרבה מקרים (בגלל האפשרויות לקיזוז שתי זרימות בכיוונים הפוכים), אבל הרעיונות המרכזים קלים יותר להבנה. בפרק זה תהיה גישה שהיא פשרה בין אלו: הזרימות יהיו חיוביות, אבל תהיה עבורן דרישה של "קיזוז" (אין זרימה חיובית בשני כיוונים "סותרים"). הנוסחאות יהיו יותר מסובכות מאשר אלו של הגישה המתמטית, אבל לא יהיה צורך בחלוקה למקרים.

רשת זרימה $N=(G,s,t,c)$ מוגדרת ע"י גרף מכוון פשוט $G=(V,E)$ (ללא לולאות או קשתות מקבילות, יכולות להיות קשתות אנטי־מקבילות), צומת מקור $s\in V$, צומת בור (יעד) $t\in V$, ופונקצית קיבול קשתות $c:E\to\mathbb{R}^+\setminus\{0\}$ (הפונקציה מקבלת ערכים ממשיים חיוביים בלבד). לכל צומת $v\in V$ נסמן ב־$\mathrm{in}(v)\subseteq V$ את קבוצת הצמתים שיש מהם קשתות נכנסות ל־v, וב־$\mathrm{out}(v)\subseteq V$ את קבוצת הצמתים שיש אליהם קשתות יוצאות מ־v.

הערה: ייתכן שגם $uv\in E$ וגם $vu\in E$ נמצאות ברשת הזרימה (והן יכולות להיות עם קיבולים שונים). בספרות לפעמים מתעלמים מאפשרות זו על מנת לפשט את ההגדרות וההוכחות. כאן לא נתעלם ממנה.

באשר למבנה הנתונים, נניח שנתונות לנו רשימות שכנויות גם עבור $\mathrm{in}(v)$ וגם עבור $\mathrm{out}(v)$, וכן שלכל קשת $uv\in E$ רשום לנו האם $vu\in E$. כך אפשר למשל לוודא האם פונקצית זרימה היא חוקית (ראו הגדרה בהמשך) בזמן $O(|V|+|E|)$. אפשר לקבל את כל הרשימות הנ"ל מתוך קלט התחלתי שנתון רק כרשימות שכנויות "רגילות" בזמן $O(|V|+|E|)$. מעתה לשם פשטות נניח גם שמתקיים $O(|V|+|E|)=O(|E|)$, הנחה שמתקיימת למשל אם אין צמתים מבודדים בגרף.

בהינתן רשת זרימה, פונקצית זרימה היא פונקציה $f:E\to\mathbb{R}^+$. אפשר לחשוב על קשתות G כ"צינורות" עם "הספק מקסימלי" שמוגדר ע"י c, ואז נחשוב על f כעל "זרימה" של נוזל מ־s ל־t דרך הגרף. על פונקצית זרימה חוקית לקיים שני תנאים:

- תנאי הקיבול (נקרא גם "חוק הקשת"): לכל $e\in E$ מתקיים $0\le f(e)\le c(e)$.
- שימור הזרימה (נקרא גם "חוק הצומת"): עבור כל צומת $u\in V\setminus\{s,t\}$ מתקיים השוויון $\sum_{v\in\mathrm{in}(u)}f(vu)=\sum_{v\in\mathrm{out}(u)}f(uv)$. אפשר לחשוב על הזרימה ככזו של "נוזל אי־דחיס".

בצמתים s ו־t אנחנו לא מחייבים את חוק הצומת, ונגדיר את עוצמת הזרימה ע"י ההפרש עבור t: נסמן $|f|=\sum_{v\in\mathrm{in}(t)}f(vt)-\sum_{v\in\mathrm{out}(t)}f(tv)$. נראה עתה שיש בעניין זה "סימטריה" בין t ל־s.

הערה והנחה להמשך: אם עבור $u,v\in V$ כל שהם מתקיים גם $uv\in E$ וגם $vu\in E$, עדיין ניתן להניח ש־$f(uv)=0$ או $f(vu)=0$ (או שניהם), מכיוון שניתן לחסר משני ערכים אלו את $\min\{f(uv),f(vu)\}$ ולהישאר עם זרימה חוקית "מקוזזת" בעלת אותה עוצמה. מעתה והלאה נניח הנחה זו.

טענה מהירה (מקור מול בור): מתקיים גם $|f|=\sum_{v\in\mathrm{out}(s)}f(sv)-\sum_{v\in\mathrm{in}(s)}f(vs)$.

הוכחה: ראשית נשים לב שלכל פונקציה $f:E\to\mathbb{R}$, בלי קשר לשאלה האם זו זרימה חוקית, מתקיים $\sum_{u\in V}\left(\sum_{v\in\mathrm{in}(u)}f(vu)-\sum_{v\in\mathrm{out}(u)}f(uv)\right)=0$, מכיוון שבסכימה הכוללת לכל $uv\in E$ צרפנו

לסכום את $f(uv)$ פעם עם "$+$" (עבור צומת הכניסה v) ופעם עם "$-$" (עבור צומת היציאה u). אבל אם f היא זרימה חוקית, אז ע"פ חוק הצומת כל האיברים בסכימה לפי v הם אפס פרט לצמתי המקור והבור, ולכן מתקיים $0 = \left(\sum_{v\in\text{in}(t)} f(vt) - \sum_{v\in\text{out}(t)} f(tv)\right) + \left(\sum_{v\in\text{in}(s)} f(vs) - \sum_{v\in\text{out}(s)} f(sv)\right)$. מכך נובע המבוקש ע"י העברת אגפים. $\blacksquare$

הגדרת בעיית הזרימה המקסימלית: בהינתן רשת זרימה, אנחנו נהיה מעוניינים בזרימה חוקית אשר משיגה את המקסימום עבור $|f|$. אפשר להראות שזה באמת מקסימום (ז"א שיש f כזו) ולא סופרימום באמצעות משפט בולצאנו־ויירשטראס (שלמדתם בחדו"א – תוך שימוש בזה שתנאי החוקיות משתמשים רק באי־שוויונים מסוג "$\leq$"), אבל בהמשך נראה אפיון קומבינטורי שמבטיח את קיום המקסימום.

לפני שנמשיך, נגדיר הרחבות של פונקצית הקיבול ופונקצית הזרימה, שיעזרו לנו לפשט את הטיעונים המתמטיים בהמשך. הרעיון הוא שניתן לחשוב על "לא־קשתות" כעל "קשתות עם קיבול 0".

הגדרה (הרחבות): עבור רשת זרימה $N = (G, s, t, c)$ המוגדרת ע"י גרף $G = (V, E)$, זוג צמתים s, t ופונקצית קיבול c, נגדיר את פונקציה הקיבול המורחבת $\tilde{c} : V \times V \to \mathbb{R}^+$ לפי $\tilde{c}(uv) = c(uv)$ אם $uv \in E$, ואחרת $\tilde{c}(uv) = 0$. באופן דומה, בהינתן זרימה $f : E \to \mathbb{R}^+$, נגדיר את ההרחבה שלה $\tilde{f} : V \times V \to \mathbb{R}^+$.

אתם מוזמנים לוודא את הטענה הבאה.

טענה (תנאי להרחבה): פונקציה נתונה $\tilde{f} : V \times V \to \mathbb{R}^+$ תהיה הרחבה של זרימה חוקית מתאימה f אם ורק אם היא מקיימת $0 \leq \tilde{f}(uv) \leq \tilde{c}(uv)$ לכל $uv \in V \times V$ וכן $\sum_{v\in V} \tilde{f}(vu) = \sum_{v\in V} \tilde{f}(uv)$ לכל $u \in V \setminus \{s, t\}$. כמו כן מתקיים אז $|f| = \sum_{v\in V} \tilde{f}(tv) - \sum_{v\in V} \tilde{f}(vt)$.

מבחינה מתמטית היה אפשר לעבוד רק עם $\tilde{c}$ ו־$\tilde{f}$ ולוותר על הגרף G, כי מתקיים $E = \{uv : \tilde{c}(uv) > 0\}$. עם זאת, אנחנו נרצה אלגוריתמים שרצים מהר יותר עבור $|E|$ קטן יותר, ולכן מבחינה אלגוריתמית נמשיך לעבוד עם ההגדרה המלאה של רשתות זרימה. עבור $uv \in V \times V$ שיופיעו אצלנו, ניתן יהיה לחשב את $\tilde{c}(uv)$ בזמן $O(1)$ בהסתמך על G ועל c.

חתכים והקשר בין חתך מינימום לזרימת מקסימום

חתך־st ברשת הזרימה שלנו יוגדר כחתך (S, T) (כזכור $S \subseteq V$ ו־$T = V \setminus S$) שמקיים בנוסף $s \in S$ ו־$t \in T$. אנחנו נסמן ב־$E(S, T)$ את הקשתות היוצאות מ־S ל־T (בשפה מתמטית אפשר לכתוב זאת "$E(S, T) = E \cap (S \times T)$"), וב־$E(T, S)$ את אלו מ־T ל־S. עתה נראה למה שמכלילה את הטענה המהירה על מקור מול בור מתת־הפרק הקודם.

למה (זרימה דרך חתך): לכל זרימה חוקית f ברשת הזרימה ולכל חתך־st (S, T) שלה מתקיים השוויון $|f| = \sum_{e\in E(S,T)} f(e) - \sum_{e\in E(T,S)} f(e)$.

הוכחה: נחשב את $\sum_{u\in T} \left(\sum_{v\in\text{in}(u)} f(vu) - \sum_{v\in\text{out}(u)} f(uv)\right)$. לכל $u \in T \setminus \{t\}$ הביטוי בסוגריים הוא 0 (לפי חוק הצומת בזרימה), ולכן נשאר לנו בסכום הכולל $|f| = \sum_{v\in\text{in}(t)} f(vt) - \sum_{v\in\text{out}(t)} f(tv)$. מצד שני, נשים לב שביטוי הסכום הנ"ל שווה ל־$\sum_{e\in E(S,T)} f(e) - \sum_{e\in E(T,S)} f(e)$. הדבר נובע מכך שכל קשת פנימית ל־T מקזזת את עצמה בסכום הזה (היא מופיעה עם "$+$" בצומת הכניסה ועם "$-$" בצומת היציאה), בעוד שכל קשת פנימית ל־S אינה מופיעה בסכום בכלל. נשארו לנו רק הקשתות מ־S ל־T, שמופיעות כל אחת בסכום פעם אחת עם סימן חיובי (יחד עם צומת הכניסה שלהן ב־T), והקשתות מ־T ל־S, שמופיעות כל אחת בסכום פעם אחת עם סימן שלילי (יחד עם צומת היציאה שלהן ב־T).

קיבלנו ששני הביטויים שווים לסכום המקורי $\sum_{u\in T}\Big(\sum_{v\in \mathrm{in}(u)} f(vu)-\sum_{v\in \mathrm{out}(u)} f(uv)\Big)$, ולכן הם שווים זה לזה. ■

נגדיר עבור החתך (S,T) את הקיבול המקסימלי שלו $c(S,T)=\sum_{e\in E(S,T)} c(e)$.

מסקנה (חסם זרימה ע"י חתך): לכל זרימה חוקית f ולכל חתך־st מתקיים $|f|\le c(S,T)$.

הוכחה: לפי למת הזרימה דרך חתך וחוק הקשת בזרימה, עבור הזרימה f והחתך (S,T) מתקיים $|f|=\sum_{e\in E(S,T)} f(e)-\sum_{e\in E(T,S)} f(e)\le\sum_{e\in E(S,T)} f(e)\le\sum_{e\in E(S,T)} c(e)=C(S,T)$. ■

מכאן אנו רואים שאפשר לחסום את גודל הזרימה החוקית המקסימלי $\max_f |f|$ ע"י קיבול החתך המינימלי $\min_{(S,T)} c(S)$. בקרוב נראה שזה לא רק חסם אלא שוויון. ההגדרה הבאה תהיה חשובה.

הגדרה (רשת זרימה שיורית): עבור רשת זרימה $N=(G,s,t,c)$ הניתנת ע"י גרף $G=(V,E)$, צמתים s,t, ופונקציה קיבול c, ובהינתן זרימה חוקית f, רשת הזרימה השיורית $N_f=(G_f,s,t,c_f)$ מוגדרת ע"י אותם צמתים V ו־s,t, יחד עם הגרף השיורי $G_f=(V,E_f)$ ופונקצית הקיבול השיורית $c_f:E\to\mathbb{R}^+$ המוגדרים באופן הבא:

- לכל $uv\in V\times V$ מגדירים $\tilde{c}_f(uv)=\tilde{c}(uv)-\tilde{f}(uv)+\tilde{f}(vu)$ (זה לא יכול להיות שלילי בגלל חוק הקשת עבור $f(uv)$).
- בהתאם מגדירים את $E_f=\{uv:\tilde{c}_f(uv)>0\}$, ו־$c_f$ מוגדרת להיות הצמצום של $\tilde{c}_f$ ל־E_f.

שימו לב שבהחלט יכולות להיות זוג קשתות אנטי־מקבילות בגרף השיורי גם אם לא היו כאלו ברשת הזרימה המקורית. הרעיון הוא שרשת הזרימה השיורית מתארת מהם השינויים שאפשר לעשות ב־f מבלי לחרוג מחוק הקשת. למשל, אם עבור u,v מסויימים התקיים $uv\in E$ אך $vu\notin E$, אז נקבל $\tilde{c}(uv)=c(uv)-f(uv)$, ז"א בכמה מותר להגדיל את הזרימה באותה קשת, וכן $\tilde{c}(vu)=f(uv)$, ז"א בכמה מותר להקטין את הזרימה בקשת uv.

כמו כן חשוב לציין שאם $uv\in E_f$ קשת בגרף השיורי, אז uv או vu (או שתיהן) קשת ב־E המקורי.

עבור שתי פונקציות זרימה מורחבות $\tilde{f}:V\times V\to\mathbb{R}^+$ ו־$\tilde{g}:V\times V\to\mathbb{R}^+$ שמקיימות את חוק הצומת, נגדיר את הסכום $\tilde{h}:V\times V\to\mathbb{R}^+$ לפי $\tilde{h}(uv)=\max\{0,\tilde{f}(uv)+\tilde{g}(uv)-\tilde{f}(vu)-\tilde{g}(vu)\}$. שימו לב שבעצם יש לנו חיבור איבר־איבר, רק שדאגנו גם "לקזז" זרימה מול זרימה בכיוון ההפוך, ותוך כדי כך דאגנו שתישאר התכונה שלכל uv או $\tilde{h}(uv)=0$ או $\tilde{h}(vu)=0$ (או שניהם). הלמה הבאה חשובה.

למה (חיבור זרימה): אם f זרימה חוקית ברשת הזרימה $N=(G,s,t,c)$, ו־g זרימה חוקית ברשת השיורית $N_f=(G_f,s,t,c_f)$, אז הסכום $\tilde{h}$ של ההרחבות $\tilde{f}$ ו־$\tilde{g}$ הוא בעצמו הרחבה של זרימה חוקית h ברשת N, ומתקיים $N_h=(N_f)_g$ (צד ימין הוא התוצאה של של חישוב הרשת השיורית N_f, ולקיחת הרשת השיורית שלה לפי g). כמו כן מתקיים $|h|=|f|+|g|$.

הוכחה: לפי הטענה על התנאי להרחבה, בשביל החוקיות מספיק להראות שמתקיים $0\le\tilde{h}(uv)\le\tilde{c}(uv)$ לכל $uv\in V\times V$ ושמתקיים $\sum_{v\in V}\tilde{h}(uv)=\sum_{v\in V}\tilde{h}(vu)$ לכל $u\in V\setminus\{s,t\}$. מתקיים $0\le\tilde{h}(uv)$ מההגדרה. כמו כן, אם $\tilde{h}(uv)=\tilde{f}(uv)+\tilde{g}(uv)-\tilde{f}(vu)-\tilde{g}(vu)$, אז נציב לתוך הביטוי הנ"ל את $\tilde{g}(uv)\le\tilde{c}_f(uv)=\tilde{c}(uv)-\tilde{f}(uv)+\tilde{f}(vu)$, ונקבל $\tilde{h}(uv)\le\tilde{c}(uv)-\tilde{g}(vu)\le\tilde{c}(uv)$ כנדרש (האופציה השניה עבור $\tilde{h}(uv)$ היא שהוא שווה ל־0).

על מנת להוכיח שמתקיים $\sum_{v\in V}\tilde{h}(uv)=\sum_{v\in V}\tilde{h}(vu)$ עבור $u\in V\setminus\{s,t\}$, ראשית שימו לב מההגדרות שלכל u,v שהם מתקיים $\tilde{h}(uv)-\tilde{h}(vu)=\tilde{f}(uv)+\tilde{g}(uv)-\tilde{f}(vu)-\tilde{g}(vu)$. לכן מתקיים

$\sum_{v\in V}\tilde{h}(uv)-\sum_{v\in V}\tilde{h}(vu)=\sum_{v\in V}\tilde{f}(uv)-\sum_{v\in V}\tilde{f}(vu)+\sum_{v\in V}\tilde{g}(uv)-\sum_{v\in V}\tilde{g}(vu)$, ואגף ימין מתאפס לפי החוקיות של f ו־g.

ההוכחה עבור $|h|=|f|+|g|$ כמעט זהה להוכחה של חוק הצומת, רק שעושים את הסכום עבור $u=t$.

לבסוף, על מנת להוכיח $N_h=(N_f)_g$, מספיק להוכיח $\tilde{c}_h=(\tilde{c}_f)_g$ (כי פונקציה זו קובעת גם את c_h וגם את G_h), וזה נובע ישירות מההגדרות של רשת שיורית (ושימוש נוסף בזה שלכל u,v מתקיים $\tilde{h}(uv)-\tilde{h}(vu)=\tilde{f}(uv)+\tilde{g}(uv)-\tilde{f}(vu)-\tilde{g}(vu)$). ■

מסלול שיפור עבור רשת הזרימה $N=(G,s,t,c)$, ביחס לזרימה חוקית f, מוגדר כמסלול מכוון מ־s ל־t בגרף השיורי G_f.

למה (שיפור זרימה): אם P הוא מסלול שיפור ביחס לזרימה f, ו־δ הוא הקיבול המינימלי לפי c_f בקשתות של P (שימו לב ש־$\delta>0$), אז ניתן למצוא זרימה חוקית h המקיימת $|h|=|f|+\delta$.

הוכחה: נסמן את המסלול $P=u_0,\ldots,u_k$ כאשר $u_0=s$ ו־$u_k=t$, ונגדיר זרימה g ברשת הזרימה השיורית ביחס ל־f באופן הבא: לכל $0<i\le k$ נגדיר $g(u_{i-1}u_i)=\delta$, ולכל קשת אחרת בגרף השיורי, $e\in E_f\setminus\{u_{i-1}u_i:0<i\le k\}$, נגדיר $g(e)=0$. זוהי זרימה חוקית: היא מקיימת את חוק הקשת כי הנחנו ש־δ הוא הערך המינימלי בקיבולי הקשתות $\{u_{i-1}u_i:0<i\le k\}$, ועבור חוק הצומת ראשית נשים לב שלכל $u\in V\setminus\{u_0,\ldots,u_k\}$ כל הזרימות $g(uv)$ ו־$g(vu)$ עבור $v\in V$ כל שהוא הן 0. עבור u_i כאשר $1<i<k$ יש רק שתי זרימות רלוונטיות שאינן 0, $g(u_{i-1}u_i)=g(u_iu_{i+1})=\delta$, והצבתן בביטוי של חוק הצומת ייתן שוויון. כמו כן $|g|=\delta$, כי בביטוי $\sum_{v\in\mathrm{in}(t)}g(vt)-\sum_{v\in\mathrm{out}(t)}g(tv)$ האיבר היחידי שאינו 0 הוא $g(u_{k-1}t)=\delta$. לפי הלמה על חיבור זרימה, נוכל אם כן להגדיר את h כחיבור של f ו־g, והוא יהיה הזרימה המבוקשת. ■

עכשיו אנחנו יכולים להוכיח את המשפט המרכזי על רשתות זרימה.

משפט (חתך מינימום וזרימת מקסימום): תהי f פונקצית זרימה חוקית ברשת $N=(G,s,t,c)$. התנאים הבאים שקולים.

1. הזרימה f היא זרימה מקסימלית.
2. אין מסלול שיפור מ־s ל־t בגרף השיורי G_f.
3. קיים חתך־st ברשת שהקיבול שלו שווה ל־$|f|$.

הוכחה: על מנת להראות את שקילות התנאים נוכיח את 2 מ־1, את 3 מ־2, ואת 1 מ־3.

טענה 2 נובעת מטענה 1: אם (בשלילה) היה מסלול שיפור P ב־G_f, אז לפי הלמה על שיפור זרימה היה אפשר למצוא זרימה חוקית h עבורה $|h|=|f|+\delta$, כאשר δ הוא המינימום של הקיבולים השיוריים (הגדולים מ־0) של קשתות P, בסתירה.

טענה 3 נובעת מטענה 2: אם אין מסלול שיפור ב־G_f, נגדיר את S להיות קבוצת כל הצמתים שאליהם יש מסלול (מכוון) מ־s ב־G_f, ואת $T=V\setminus S$ כקבוצת הצמתים אליהם אין מסלול כזה. לפי ההנחה שאין מסלול שיפור, (S,T) הוא חתך־st, ומההגדרה שלו נובע שב־G_f אין קשתות מ־S ל־T (מותר שיהיו קשתות בכיוון ההפוך). אם מסתכלים על הגדרת G_f לפי רשת הזרימה המקורית ולפי f, נובע מכך שלכל קשת $e\in E(S,T)$ שיוצאת ב־G מ־S ל־T מתקיים $f(e)=c(e)$, וכן שלכל קשת $e\in E(T,S)$ מתקיים $f(e)=0$. על כן ע"פ הלמה על זרימה לפי חתך מתקיים $c(S,T)=\sum_{e\in E(S,T)}c(e)-0=|f|$.

טענה 1 נובעת מטענה 3: זה נובע מהמסקנה מקודם על חסם זרימה לפי חתך. ■

אלגוריתם לזרימת מקסימום

לאור המשפט על חתך מינימום וזרימת מקסימום, ננסח אלגוריתם גנרי שכל פעם מנסה לחפש מסלול שיפור. זהו האלגוריתם של פורד־פאלקרסון Ford-Fulkerson.

האלגוריתם הגנרי של פורד־פאלקרסון לזרימת מקסימום

קלט: רשת זרימה $N = (G, s, t, c)$ מעל הגרף המכוון $G = (V, E)$

פלט: זרימת מקסימום $f : E \to \mathbb{R}^+$

- **אתחול:** מציבים $f(e) \leftarrow 0$ לכל $e \in E$
- **כל עוד** יש מסלול שיפור ברשת השיורית N_f
  - מציבים ב־f את הזרימה המשופרת לפי למת שיפור הזרימה

טענה מיידית (נכונות בעצירה): אם וכאשר האלגוריתם עוצר, הפלט יהיה זרימה מקסימלית לפי המשפט על זרימת מקסימום וחתך מינימום. ■

באשר לזמן ריצה, אפשר לחשב את הזרימה המשופרת ואת הגרף השיורי המעודכן בזמן $O(|E|)$, ואפשר בזמן כזה גם למצוא מסלול שיפור באמצעות אחד מאלגוריתמי החיפוש המוכרים לנו. עם זאת, ללא מידע נוסף אין לנו כל הבטחה על מספר האיטרציות המקסימלי. ישנן דוגמאות עם קיבולים אי־רציונלים, ומסלולי שיפור ש"נבחרים ע"י האויב", שיביאו לסדרה אינסופית של שיפורים, ואפילו סדרה כזו שמתכנסת לערך נמוך יותר מזה של זרימת המקסימום.

אם הקיבולים רציונלים, אז כל שיפור מעלה את ערך הזרימה בלפחות $1/k$, כאשר k הוא המכנה המשותף הנמוך לכל השברים המופיעים בקיבולים, כך שנעצור בסופו של דבר (כי ערך זרימת המקסימום הוא בוודאי לא יותר מסכום כל הקיבולים בגרף).

עבור קיבולים שלמים, מספר האיטרציות לא יעלה על ערך זרימת המקסימום עצמה (כל מסלול משפר את העוצמה של הזרימה ב־1 לפחות), אולם יש מקרים שבהם הוא יהיה שווה לה. למשל, עבור הרשת עם $V = \{s, a, b, t\}$, כאשר יש קשתות בקיבול k מ־s ל־a ו־b, קשתות בקיבול k מ־a ו־b ל־t, וקשת בקיבול 1 מ־a ל־b, אז זרימת המקסימום היא בגודל $2k$. עם זאת, אפשר להריץ את האלגוריתם של פורד־פאלקרסון כך שבכל פעם לוקחים מסלול שיורי שעובר דרך ab או דרך ba (תלוי בגרף השיורי באותו רגע), ולהגיע לזרימת מקסימום תוך $2k$ איטרציות. אורך הקלט, גם אם מחשבים לתוכו את הייצוגים המספריים של המשקלים (בייצוג בינארי), יהיה $O(|E| \log(k))$ בלבד, ולכן זה אינו יכול להחשב לאלגוריתם בעל זמן ריצה פולינומי.

מייד נראה איך אפשר להפוך את האלגוריתם לפולינומי ע"י הוספת קריטריון בחירה של המסלול המשפר בכל שלב. אבל לפני כן נראה מסקנה מתמטית, אשר בהמשך תשרת אותנו גם עבור פיתוח אלגוריתמים לבעיות ספציפיות.

מסקנה מתמטית מיידית (זרימה בשלמים): אם פונקצית הקיבול של רשת הזרימה מקבלת ערכים שלמים בלבד, אז יש גם זרימת מקסימום שכל ערכיה הם שלמים בלבד, מכיוון שבפרט ההפעלה של אלגוריתם פורד־פאלקרסון תתן זרימה כזו. ■

בחזרה לעניין זמן הריצה, איך נבחר מסלול שיפור אשר יניב הגבלה על מספר האיטרציות של הלולאה המרכזית? מפתה לבחור מסלול שנותן שיפור מקסימלי, ז"א מסלול שנותן מקסימום בקיבול השיורי של הקשת המינימלית בתוכו. אפשר כך לקבל אלגוריתם שעבור קיבולים שלמים רץ בזמן ריצה פולינומי ב־$|E|$ ו־$\log(k)$, כאשר k הוא ערך הזרימה המקסימלית. אלגוריתם כזה נקרא "פולינומי־חלש".

אנחנו נהיה מעוניינים באלגוריתם פולינומי־חזק, ז"א אלגוריתם אשר מבצע מספר פולינומי ב־$|E|$ בלבד של פעולות אריתמטיות (כך שאין תלות באורך ייצוג המספרים, אלא רק זה המגולם במימוש של הפעולות האריתמטיות עצמן – חיבור, חיסור והשוואה).

האלגוריתם הבא, אלגוריתם אדמונדס־קארפ Edmonds-Karp, הוא יישום של אלגוריתם פורד־פאלקרסון עם קריטריון פשוט ביותר לבחירת המסלול המשפר, לפי אורך (מספר הקשתות) המסלול בלבד.

האלגוריתם של אדמונדס־קארפ לזרימת מקסימום

קלט: רשת זרימה $N = (G, s, t, c)$ מעל הגרף המכוון $G = (V, E)$

פלט: זרימת מקסימום $f : E \to \mathbb{R}^+$

- **אתחול:** מציבים $f(e) \leftarrow 0$ לכל $e \in E$
- **כל עוד** יש מסלול שיפור ברשת השיורית N_f
  - יהי P מסלול כזה קצר ביותר, ז"א עם מספר מינימלי של קשתות
  - מציבים ב־f את הזרימה המשופרת באמצעות P, לפי למת שיפור הזרימה

זמן כל איטרציה הוא עדיין $O(|E|)$, כאשר מבצעים חיפוש לרוחב (BFS) מ־s על G_f על מנת למצוא מסלול שיפור עם מספר קשתות מינימלי. על מנת לחסום את מספר האיטרציות הכולל, נגדיר מושג של "קשת קריטית" עבור מסלול שיפור. לאחר שנראה שכל קשת (ב־G או בגרף השיורי) יכולה להיות עם סטטוס כזה לא יותר מ־$O(|V|)$ פעמים בכל מהלך האלגוריתם, זה יתן לנו חסם של $O(|V||E|)$ על מספר האיטרציות, ולכן חסם זמן כולל של $O(|V||E|^2)$ על האלגוריתם המלא. בפרט זה יהיה אלגוריתם פולינומי חזק.

כאשר נתונה רשת $N = (G, s, t, c)$ וזרימה f, נסמן לכל $v \in V$ ב־$d_f(v)$ את אורך המסלול הקצר ביותר (מספר קשתות מועט ביותר) בגרף השיורי G_f מ־s ל־v (צמתים שאין עבורם מסלול כזה כלל לא יהיו רלוונטים עבור המשך הניתוח). בנוסף, בהינתן מסלול שיפור P עבור f, נגיד שקשת e במסלול היא קריטית אם $c_f(e)$ מקבל ערך מינימלי מבין הקיבולים השיוריים של קשתות המסלול.

הטענות הבאות מתיחסות לרשת N, זרימה f, מסלול שיפור P עבור f שנתון עליו שהוא קצר (בעל מספר קשתות מועט) ביותר, והזרימה h המתקבלת באמצעות שיפור f ע"י P.

טענה (מחיקת קשת): אם e היא קשת קריטית במסלול השיפור P עבור f, אז e אינה קשת בגרף השיורי G_h, כאשר h היא הזרימה המתקבלת לאחר שיפור f לפי P.

הוכחה: מכיוון שזו קשת קריטית, מתקיים $c_f(e) = \delta$, כאשר δ הוא הערך שבו משתמשים להגדרת הזרימה g (לפי המסלול P) שאותה מחברים ל־f (ראו את הגדרת הזרימה הנ"ל בהוכחת הלמה על שיפור זרימה). נסמן ב־e' את זוג הצמתים האנטי־מקביל ל־e. לפי הוכחת הלמה על חיבור זרימה מתקיים $\tilde{c}_h = (\tilde{c}_f)_g$, ולכן $\tilde{c}_h(e) = \tilde{c}_f(e) - \tilde{f}(e) + \tilde{f}(e') = \delta - \delta + 0 = 0$, ז"א ש־$e$ אינה נמצאת ב־E_h. ■

טענה (הוספת קשת): אם $e = uv$ נוספה ל־E_h אבל לא היתה ב־E_f, אז $d_f(u) - d_f(v) = 1$.

הוכחה: אם uv נוספה ל־E_h המתקבל משיפור f לפי המסלול הקצר ביותר P, אז בהכרח vu היא קשת ב־P (כי vu חייבת לקבל זרימה חיובית ב־g כדי ש־uv שלא היתה ב־E_f תוכנס ל־E_h – שימו לב שהיה כאן עוד שימוש נסתר בלמה על חיבור זרימות). עתה ניזכר בטענה על תתי־מסלולים קלים מהפרק על מסלולים קלים ביותר. ההגדרות של אורכי מסלולים כאן מתאימות למקרה פרטי של מסלולים קלים ביותר, כאשר פונקצית המשקל היא 1 על כל הקשתות. זה אומר שגם תת־המסלול של P מ־s ל־u, וגם תת־המסלול של P מ־s ל־v, שניהם מסלולים קצרים ביותר לצמתים הנ"ל. על כן $d_f(u) - d_f(v) = 1$, לפי המיקום היחסי של u ו־v ב־P. ■

החשוב בטענה הבאה הוא שאפשר להגיד משהו על המרחק d_f לפני השיפור, ביחס לקשת $uv \in G_h$ הנמצאת בגרף השיורי שלאחר השיפור (גם כאן, הטענה מתיחסות לשיפור ע"י מסלול שיפור קצר ביותר).

טענה (הפרש מרחקים דרך קשת): אם $uv \in E_h$, אז $d_f(v) - d_f(u) \leq 1$.

הוכחה: אם מתקיים גם $uv \in E_f$, אז זוהי תכונה בסיסית של מרחקים שנובעת מאי־שוויון המשולש. אם לעומת זאת מתקיים $uv \notin E_f$, אז משתמשים בטענה על הוספת קשת לקבלת $d_f(v) - d_f(u) = -1$, וכידוע $-1 \leq 1$. ■

מהטענה האחרונה נובעת למה מרכזית עבור שיפור ע"י מסלול קצר ביותר.

למה (מונוטוניות דרך שיפורים): אם h היא שיפור של f ע"י מסלול שיפור קצר ביותר, אז $d_h(v) \geq d_f(v)$ לכל $v \in V$.

הוכחה: נסמן $d_h(v) = k$, ונסמן מסלול ב־G_h המשיג את המרחק הזה ב־$u_0, \ldots, u_k$ כאשר $u_0 = s$ ו־$u_k = v$. מתקבל $d_h(v) = \sum_{i=1}^{k} 1 \geq \sum_{i=1}^{k}(d_f(u_i) - d_f(u_{i-1})) = d_f(u_k) - d_f(u_0) = d_f(v)$ לפי הלמה על הפרש מרחקים דרך קשת (השוויון השני מימין הוא של סכימה טלסקופית). ■

עכשיו ניתן לחסום את מספר הפעמים שקשת (בגרף G_f) היא קריטית.

למה (חסם קריטיות): זוג צמתים uv (כאשר $uv \in E$ ו/או $vu \in E$) יהווה קשת קריטית במסלול שיפור במהלך ביצוע האלגוריתם אדמונדס־קארפ לא יותר מ־$|V|/2$ פעמים.

הוכחה: נסמן ב־$f_0, \ldots, f_\ell$ את הזרימות במהלך ההרצה של האלגוריתם, כאשר f_0 היא זרימת ה־0 שאיתה האלגוריתם התחיל, ו־f_ℓ היא זרימת המקסימום הסופי. לפי הלמה על מונוטוניות דרך שיפורים, $d_{f_0}(u) \leq d_{f_1}(u) \leq \cdots \leq d_{f_\ell}(u)$.

אם uv קריטית במסלול המשפר ל־f_i מסויים, אז לפי הלמה על מחיקת קשת uv אינה קשת ב־$E_{f_{i+1}}$. כמו כן $d_{f_i}(v) = d_{f_i}(u) + 1$ כי זוהי קשת על מסלול קצר ביותר. אם uv חוזרת בפעם הבאה להיות קשת ב־E_{f_j} (עבור ה־$j > i$ המינימלי המתאים), אז לפי הלמה על הוספת קשת $d_{f_{j-1}}(u) = d_{f_{j-1}}(v) + 1$, ולפי הלמה על מונוטוניות נובע משני אי־השוויונות האחרונים $d_{f_j}(u) \geq d_{f_i}(u) + 2$. מכיוון שהמרחק $d_f(u)$ תמיד חסום ע"י $|V|$, נובע מכך שהזוג uv אינו יכול להיות קשת קריטית יותר מ־$|V|/2$ פעמים. ■

נעיר שלמרות שמהסתכלות על ההוכחה מפתה לנסות להראות ש־u עצמו אינו יכול להיות מוצא של קשת קריטית יותר מ־$|V|/2$ פעמים, טענה כזו אינה נכונה. נסו לראות באיזה שלב נסיון הוכחה כזה יכשל.

ניגש עתה להוכחה של המשפט המרכזי על סיבוכיות האלגוריתם של אדמונדס־קארפ.

משפט (סיבוכיות אדמונדס־קארפ): מספר השיפורים הכולל בהרצה של אלגוריתם אדמונדס־קארפ אינו עולה על $|V||E|$, ולכן הסיבוכיות הכוללת של האלגוריתם חסומה ע"י $O(|V||E|^2)$.

הוכחה: ישנם לא יותר מ־$2|E|$ זוגות uv שיכולים להיות קשת בגרף שיורי כל שהוא (כי כזכור זוג כזה חייב לקיים $uv \in E$ ו/או $vu \in E$). מכיוון שבכל איטרציה לפחות אחד הזוגות המעורבים הוא קריטי, ומצד שני אף זוג אינו קריטי יותר מ־$|V|/2$ פעמים לפי הלמה על חסם הקריטיות, מתקבל החסם המבוקש על מספר השיפורים הכולל.

עבור חסם הזמן, נשים לב שביצוע חיפוש לרוחב עבור מציאת מסלול השיפור הקצר ביותר לוקח זמן $O(|E|)$ (כזכור אנחנו מניחים $O(|V| + |E|) = O(|E|)$ במהלך הפרק), והעדכון לפי המסלול שמצאנו (שאורכו לכל היותר $|V|$) לוקח זמן $O(|V|)$. סה"כ לוקח לנו זמן $O(|E|)$ לכל מסלול שיפור, ובסה"כ $O(|V||E|^2)$. זמן האתחול של האלגוריתם, $O(|E|)$, נבלע בזמן זה. ■

היישום לשידוכים בגרף דו־צדדי

שידוך (קרוי גם זיווג) בגרף לא מכוון $G = (V, E)$ הוא קבוצה של קשתות $M \subseteq E$ זרות זו לזו, ז"א שאין צומת המשתתף ביותר מקשת אחת. שידוך מושלם הוא שידוך המכסה את כל קבוצת הצמתים (בפרט שידוך מושלם ייתכן רק אם $|V|$ זוגי). גרף דו־צדדי $G = (L, R, E)$ הוא גרף שבו נתונה מראש חלוקה של קבוצת הצמתים לשתי קבוצות זרות, $V = L \cup R$ כאשר $L \cap R = \emptyset$, וקבוצת הקשתות E כוללת רק קשתות עם צומת אחד מ־L וצומת אחד מ־R.

אנחנו נדון בפרק זה ביישום של רשתות זרימה למציאת שידוכים בעלי מספר קשתות מקסימלי בגרפים דו־צדדיים, וכן במסקנות קומבינטוריות שניתן להסיק מעבר ליכולת למצוא שידוכים אלו ביעילות.

אנחנו נשתמש כאן בבניה הבאה:

בניה של רשת זרימה למציאת שידוך מקסימלי בגרף דו־צדדי

קלט: גרף דו־צדדי פשוט $G = (L, R, E)$

פלט: רשת זרימה N שעוצמת הזרימה המקסימלית בה שווה לגודל השידוך המקסימלי ב־G

- מגדירים את הגרף המכוון $G' = (V', E')$ באופן הבא:
  - מגדירים $V' = L \cup R \cup \{s, t\}$, כאשר s ו־t הם צמתים חדשים שלא היו בגרף המקורי
  - מגדירים את $E' = \{su : u \in L\} \cup \{wt : w \in R\} \cup \{uw : uw \in E,\, u \in L,\, w \in R\}$ (שימו לב לכיוונים של הקשתות בקבוצה)
- מגדירים את $c : E' \to \mathbb{R}^+ \setminus \{0\}$ לפי $c(e) = 1$ לכל $e \in E'$

- רשת הזרימה היא $N=(G',s,t,c)$

נראה עתה את הקשר של רשת הזרימה הנ"ל לשידוכים בגרף

למה (משידוך לזרימה): אם יש ב־G שידוך M בעל k קשתות, אז יש ב־N זרימה בשלמים בעוצמה k.

הוכחה: לכל $uw\in M$ כאשר $u\in L$ ו־$w\in R$ נקבע $f(su)=f(uw)=f(wt)=1$. לכל קשת $e\in E'$ פרט לאלו שכבר קבענו עבורן את f, נקבע $f(e)=0$. ברור שכלל הקשת מתקיים. אם הצומת $u\in L$ אינו מוכל באף קשת של M, מתקיים $f(su)=0$ וכן לכל $uw\in E'$ מתקיים $f(uw)=0$, כך שמתקיים עבורו כלל הצומת. אם $u\in L$ מוכל בקשת של M, אז הוא מוכל בקשת אחת uw כזו בדיוק (כי זהו שידוך), ויתקיים $f(su)=f(uw)=1$, כאשר בשאר הקשתות של u הזרימה היא 0 ולכן כלל הצומת מתקיים.

אתם מוזמנים להשלים את ההוכחה שכלל הצומת מתקיים גם עבור צמתי R (בעצם אותה הוכחה בדיוק), וכן שעוצמת הזרימה היא $|M|$, למשל ע"י חישוב הזרימה בחתך $(\{s\}\cup L,R\cup\{t\})$. ■

למה (מזרימה לשידוך): אם יש ב־N זרימה חוקית בשלמים f, אז יש ב־G שידוך בעל $|f|$ קשתות.

הוכחה: מכיוון שכל הקיבולות הן 1, זרימה בשלמים פירושה שלכל $e\in E'$ מתקיים $f(e)=0$ או $f(e)=1$. נבחר את M להיות קבוצת הקשתות ב־G (בין L ל־R) שעבורן f מקבלת ערך 1. מספר הקשתות הנ"ל שווה לעוצמה של f, לפי הזרימה דרך החתך $(\{s\}\cup L,R\cup\{t\})$. לא יכול להיות צומת ב־L שמשתתף ביותר מקשת אחת ב־M, כי הזרימה הנכנסת לכל $u\in L$ חסומה ע"י 1 (יש קשת נכנסת בודדת). באופן דומה מראים שאין צומת ב־R שמשתתף ביותר מקשת אחת ב־M, כך ש־M הוא אכן שידוך. ■

המסקנה עבור מציאה אלגוריתמית של שידוך מקסימלי היא מיידית.

מסקנה מיידית (מציאת שידוך מקסימלי): מכיוון שברשת זרימה עם משקלות שלמים תמיד יש זרימה מקסימלית בשלמים, ומכיוון שאלגוריתם אדמונדס־קארפ מוצא זרימה כזו, קיים אלגוריתם שבהינתן גרף דו־צדדי $G=(L,R,E)$, מוצא בזמן פולינומי שידוך מקסימלי ב־G. ■

עתה נדון בתוצאות קומבינטוריות של הבניה ושתי הלמות המקשרות בין שידוך וזרימה. אנחנו נהיה מעוניינים בגרף דו צדדי $G=(L,R,E)$ שבו $|L|=|R|=n$, ונרצה למצוא תנאים לקיום שידוך מושלם (שבו מספר הקשתות גם יהיה n). ראשית, נראה מסקנה מהירה שהיה קשה להוכיח ללא המעבר לרשתות זרימה.

מסקנה (שידוך בשוויון דרגה): אם כל צמתי G בעלי אותה דרגה (מספר שכנים), אז קיים שידוך מושלם.

הוכחה: מספיק לנו להראות שקיימת זרימה חוקית f בעלת עוצמה n ב־G שאינה בשלמים. מכך נובע שקיימת גם זרימה f' בעלת אותה עוצמה שהיא בשלמים, וזו נותנת את השידוך.

על מנת לבנות את f, נסמן ב־k את הדרגה המשותפת לצמתים. עבור כל קשת ב־G' מהצורה su נקבע $f(su)=1$, עבור כל קשת מהצורה wt גם נקבע $f(wt)=1$, ועבור כל קשת uw כאשר $u\in L$ ו־$w\in R$ נקבע $f(uw)=1/k$. לא קשה לוודא שזרימה זו אכן מקיימת את התנאים הנדרשים. ■

נשתמש עתה ברשתות זרימה על מנת להוכיח את המשפט המפורסם הבא:

משפט הול Hall על שידוכים בגרף דו־צדדי: אם $G=(L,R,E)$ מקיים $|L|=|R|$, וכן שלכל קבוצה $U\subseteq L$ קבוצת השכנים שלה $N(U)=\bigcup_{u\in U}\{w\in R: uw\in E\}$ מקיימת $|U|\le|N(U)|$, אז קיים שידוך מושלם ב־G.

לפני שנמשיך, שימו לב שזה בעצם "אם ורק אם". אם יש שידוך מושלם $M \subseteq E$ ב־G, אז בפרט $|M| = |R| = |L|$, וכן לכל U קבוצת השכנים שלה $N(U)$ כוללת לפחות את כל הצמתים המשודכים לצמתי U לפי M, המהווים $|U|$ צמתים.

על מנת להוכיח את המשפט, נראה קודם כל את התכונה הבאה של חתכים עם קיבול מינימלי ברשת הזרימה $N = (G', s, t, c)$ שבנינו עבור G.

טענה (חתכים ושכנויות): המינימום של הקיבול $c(S,T)$ עבור חתכי־st ברשת $N = (G', s, t, c)$ מתקבל ע"י חתך (S,T) שבו S היא מהצורה $\{s\} \cup U \cup N(U)$ כאשר $U \subseteq L$.

הוכחה: עבור החתך (S,T) נסמן $U = S \cap L$ ו־$W = S \cap R$. נראה שאם $W \neq N(U)$, אז אפשר לשנות את החתך ולהגיע לחתך־st אחר (S',T') שבו כן מתקיים $W' = N(U')$ עבור ההגדרות המתאימות של $U' = S' \cap L$ ו־$W' = S' \cap R$. נעשה את השינוי בשלבים, כשכל פעם נזיז צומת בודד.

אם קיים צומת $w \in R \setminus S$ שהוא שכן של צומת $u \in S \cap L$, נראה שאם מעבירים את w מ־T ל־S הקיבול של החתך לא גדל. בהעברה כזו נוספת רק קשת בודדת מ־S ל־T, הקשת wt. הסיבה לכך היא שאין מ־w קשתות יוצאות אחרות, וקשתות נכנסות ל־w לא יספרו (הן יהיו "בכיוון ההפוך" לצורך חישוב קיבול החתך). מצד שני, בהכנסת w ל־S החתך איבד לפחות קשת אחת, את הקשת uw. מכיוון שלכל הקשתות קיבול 1, יוצא שקיבול החתך לא גדל.

מצד שני, אם קיים צומת $w \in R \cap S$ שאינו שכן של אף צומת מ־$S \cap L$, נראה שקיבול החתך לא יגדל אם נעביר אותו מ־S ל־T. הסיבה לכך היא שלפי הנתון אין ל־w אף קשת נכנסת מצומת אחר של S (אין לו קשת מ־s כי הוא שייך ל־R, אין לו קשת מ־$S \cap L$ לפי הנתון, ואין קשתות כלל בתוך R), כך שהעברתו ל־T לא מוסיפה כלל קשתות לחתך.

לאחר שעושים את כל ההעברות מקבלים חתך (S',T') שעבורו $S' \cap R = N(S' \cap L)$, והוא בעל הצורה הנדרשת עבור $U = S' \cap L$. ■

מכאן אפשר להוכיח את משפט הול.

הוכחה של משפט הול: נראה שקיבול החתך המינימלי הוא $|L| = |R| = n$, ומכאן נקבל שיש זרימה בעוצמה n (ולכן גם שידוך מושלם) מהמשפט על זרימה מקסימלית וקיבול מינימלי.

לפי הטענה על חתכים ושכנויות מספיק לנו לבחון חתכים (S,T) שבהם S הוא מהצורה $\{s\} \cup U \cup N(U)$ כאשר $U \subseteq L$. הקשתות בחתך כזה הן $n - |U|$ הקשתות מ־s לאיברי $L \cap T = L \setminus U$, בתוספת $|N(U)| \geq |U|$ הקשתות מאיברי $S \cap R = N(U)$ ל־t. סה"כ לפחות n קשתות. במקרה של $S = \{s\}$ קיבול החתך הוא בדיוק n, ולכן זהו קיבול החתך המינימלי. ■

זרימות עם חסמים תחתונים

לפעמים נתקלים בבעיה, שכאשר מעבירים אותה למודל זרימה, נדרשים גם לקיום זרימה מינימלית בקשתות מסויימות, ולא רק לחסם על זרימה מקסימלית. באופן פורמלי, רשת זרימה עם חסמים תחתונים נתונה ע"י $N = (G, s, t, c, b)$, כאשר G גרף מכוון (כמו ברשת זרימה רגילה), s ו־t צמתים של G, $c : E \to \mathbb{R}^+ \setminus \{0\}$ פונקצית קיבול, ו־$b : E \to \mathbb{R}^+$ פונקצית חסם המקיימת $b(e) \leq c(e)$ לכל $e \in E$. יכולות להיות קשתות שעבורן $b(e) = 0$, כלומר קשתות שעבורן אין דרישה של חסם תחתון.

זרימה חוקית $f : E \to \mathbb{R}^+$ חייבת לקיים את כלל הצומת וכלל הקשת של זרימה חוקית ברשת זרימה רגילה, ובנוסף לכך עליה לקיים את תנאי החסם $f(e) \geq b(e)$ לכל $e \in E$. חשוב להדגיש כאן שברשת

זרימה עם חסמים תחתונים לא תמיד קיימת זרימה חוקית כלל, בניגוד לרשת זרימה רגילה שבה זרימת האפס תמיד חוקית. על כן, קודם כל נראה שיטה לברר האם יש זרימה חוקית כל שהיא ברשת עם חסמים תחתונים.

הערה: כשיש חסמים תחתונים, יכול להיות שזרימה חוקית חייבת לקיים $f(uv) > 0$ ו־$f(vu) > 0$, במידה ושתי הקשתות נמצאות בגרף ויש חסם תחתון שונה מ־0 על לפחות אחת מהן. אפשר במקום זאת למצוא את הזרימה ה"מקווזת", ז"א את f' שנותנת את ההפרש בין $f(uv)$ ל־$f(vu)$ במקום את הערכים המקוריים. הזרימה המקווזת מ־u ל־v חייבת להיות בין $\max\{0, b(uv) - c(vu)\}$ לבין $\max\{0, c(uv) - b(vu)\}$, והזרימה מ־v ל־u בין $\max\{0, b(vu) - c(uv)\}$ לבין $\max\{0, c(vu) - b(uv)\}$. כאשר רושמים את רשת הזרימה המתאימה לתנאים האלו (ומסירים קשתות שהופכות להיות עם קיבולת של 0), היא תקיים שלא יתכנו בין u ל־v שתי קשתות אנטי־מקבילות אלא אם כן אין עבורם חסמים תחתונים גדולים מ־0.

עבור פתרון השאלה של קיום זרימה חוקית ברשת עם חסמים תחתונים $N = (G, s, t, c, b)$, נבנה רשת זרימה רגילה $N' = (G', s', t', c')$ שבה תהיה זרימה עם עוצמה מסויימת רק אם ברשת המקורית היתה זרימה חוקית. הבניה נעשית באופן הבא.

בניה של רשת זרימה רגילה עבור רשת עם חסמים תחתונים

קלט: רשת זרימה עם חסמים תחתונים $N = (G, s, t, c, b)$

פלט: רשת זרימה (רגילה) N', שעוצמת הזרימה המקסימלית בה מעידה על קיום של זרימה חוקית ב־N

- מגדירים $V' = V \cup \{s', t'\}$, כאשר s' ו־t' הם שני צמתים חדשים שלא היו ברשת המקורית
- מגדירים את $E'' = E \cup \{ts\} \cup \{s'u : u \in V\} \cup \{ut' : u \in V\}$ (אחר כך נסיר מ־E'' עוד קשתות עם קיבול 0)
- מגדירים את $c'' : E'' \to \mathbb{R}^+$ באופן הבא:
  - **לכל** $e \in E$ מגדירים $c''(e) = c(e) - b(e)$
  - **לכל** $u \in V$ מגדירים $c''(s'u) = \sum_{v \in \text{in}(u)} b(vu)$ ו־$c''(ut') = \sum_{v \in \text{out}(u)} b(uv)$
  - מגדירים $c''(ts) = +\infty$
- מגדירים את $E' = \{e \in E'' : c''(e) > 0\}$, את c' להיות הצמצום של c'' ל־E', ואת $G' = (V', E')$.
- רשת הזרימה היא $N' = (G', s', t', c')$

משפט (תנאי לקיום זרימה עם חסמים): קיימת זרימה חוקית ב־N אם ורק אם עוצמת הזרימה המקסימלית ב־N' היא $\sum_{e \in E} b(e)$.

הוכחה: בכל מקרה, עוצמת הזרימה המקסימלית אינה יכולה לעלות על $\sum_{v \in \text{in}'(t')} c'(vt') = \sum_{e \in E} b(e)$. שימו לב שאנחנו מסמנים ב־$\text{in}'(u)$ ו־$\text{out}'(u)$ את הקבוצות המתאימות ב־G'.

אם יש זרימה f' עם עוצמה כזו, נשים לב שבפרט מתקיים $f'(s'u) = c'(s'u) = \sum_{v\in\mathrm{in}(u)} b(vu)$ ו־$f'(ut') = c'(ut') = \sum_{v\in\mathrm{out}(u)} b(uv)$ בכל הקשתות מהצורות הנ"ל. נגדיר את $f : E \to \mathbb{R}^+$ לפי $f(e) = f'(e) + b(e)$ (אם $e \in E \setminus E'$ אז פשוט $f(e) = b(e)$). עבור $u \in V \setminus \{s,t\}$ מתקיים $\sum_{v\in\mathrm{in}'(u)} f'(vu) = \sum_{v\in\mathrm{in}(u)} f(vu) - \sum_{v\in\mathrm{in}(u)} b(vu) + f'(s'u) = \sum_{v\in\mathrm{in}(u)} f(vu)$. באופן דומה מתקיים $\sum_{v\in\mathrm{out}'(u)} f'(uv) = \sum_{v\in\mathrm{out}(u)} f(uv)$. לכן קיום חוק הצומת של f נובע ישירות מזה של f'. כלל הקשת (החסם העליון) קל לווידוא מההגדרה של $c'(uv)$, והחסם התחתון מיידי מהגדרת $f(uv)$.

מצד שני, אם יש זרימה חוקית f עבור N, אז מגדירים $f'(e) = f(e) - b(e)$ עבור $e \in E' \cap E$, את $f'(ts) = |f|$ (הגדרה זו מבטיחה את חוק הצומת עבור s ו־t, שהם צמתים "רגילים" ב־V'), ואת $f'(s'u) = c'(s'u) = \sum_{v\in\mathrm{in}(u)} b(vu)$ ו־$f'(ut') = c'(ut') = \sum_{v\in\mathrm{out}(u)} b(uv)$ בקשתות מהצורה הנ"ל. ווידוא חוק הצומת של f' עבור הצמתים הנותרים ב־$V' \setminus \{s',t'\}$ נעשה בדומה לדיון למעלה במעבר בכיוון ההפוך, וקיום חוק הקשת והדרישה על $|f'|$ שוב נובעים מההגדרות ומושארים כתרגיל. ■

עתה שאנחנו יודעים איך למצוא זרימה חוקית f ב־N אם יש כזו, איך מוצאים את הזרימה המקסימלית? לשם כך בונים "רשת שיורית" $N_f = (G_f, s, t, c_f)$, שעבור זרימה חוקית f לא תהיה בעלת חסמים תחתונים. לשם כך לכל $uv \in V \times V$ מגדירים $\tilde{c}_f(uv) = \tilde{c}(uv) - \tilde{f}(uv) + \tilde{f}(vu) - \tilde{b}(vu)$, ומ־$\tilde{c}_f$ מגדירים את E_f ו־c_f בצורה הרגילה. שימו לב שההבדל היחיד של הגדרה זו מהגדרת הרשת השיורית לרשתות ללא חסמים תחתונים הוא חיסור תנאי המינימום מהזרימה בכיוון ההפוך. ההוכחה שההגדרה עובדת (בעצם ההרחבה של הלמה על חיבור זרימה למקרה כאן) דומה למה שנעשה עבור רשתות ללא חסמים תחתונים, רק עם יותר טירחה.

אלגוריתם משופר לזרימה מקסימלית

נראה כאן אפשרות לייעל את האלגוריתם המוצא זרימה מקסימלית ברשת זרימה נתונה, עם סיבוכיות זמן $O(|V|^2|E|)$, במקום $O(|V||E|^2)$ של אדמונדס־קארפ. זהו האלגוריתם של דיניץ Dinitz. הרעיון בו הוא לבצע שיפור זרימה דרך הרבה מסלולים קצרים ביותר בבת אחת, במקום רק מסלול אחד (גם כאן "מרחק" ו"מסלול קצר ביותר" מתייחסים למספר הקשתות במסלולים). ראשית נגדיר את "גרף המסלולים הקצרים ביותר" ונראה איך מוצאים אותו, ואז נגדיר מה זה אומר להיות "זרימה מקסימלית" בו.

הגדרה: עבור גרף מכוון $G = (V,E)$ וצומת התחלה $s \in V$, גרף השכבות הוא הגרף המכיל את כל הקשתות הנמצאות על מסלולים קצרים ביותר מ־s לצמתים האחרים (שימו לב שזה בד"כ לא עץ). השכבות עצמן מוגדרות באמצעות קטלוג הצמתים לפי מרחקם מ־s, $V_i = \{u \in V : \mathrm{dist}(s,u) = i\}$.

טענה (קשתות ומסלולים בגרף שכבות): קשתות גרף השכבות יהיו בדיוק כל הקשתות של G מצומת ב־V_{i-1} לצומת ב־V_i עבור i כל שהוא. כמו כן, כל מסלול (מכוון) מ־s על גרף השכבות הוא בפרט מסלול קצר ביותר (ב־G) מ־s לצומת היעד שלו.

הוכחה: החלק הראשון נכון באופן מיידי עבור $i = 0$ (אין שכבה "V_{-1}", וגם ברור שאין קשתות נכנסות ל־$V_0 = \{s\}$, כי המסלול הקצר ביותר היחידי ל־s הוא זה המכיל את s בלבד).

בשביל ההוכחה עבור $i > 0$, ראשית נשים לב שכל קשת מ־$v_{i-1} \in V_{i-1}$ ל־$v_i \in V_i$ נותנת מסלול באורך i (ולכן קצר ביותר) מ־s ל־v_i: לפי ההנחות יש מסלול מאורך $i-1$ מ־s ל־v_{i-1}, ואז אפשר לשרשר לו את הקשת ל־v_i לקבלת המסלול המבוקש.

מצד שני, אם $v_{i-1}v_i$ קשת בגרף השכבות, ו־$s, v_1, \ldots, v_{i-1}v_i$ הוא המסלול הקצר ביותר ל־v_i שבזכותו הקשת נמצאת בגרף השכבות, אז תת־המסלול $s, v_1, \ldots, v_{i-1}$ הוא מסלול קצר ביותר כתת־מסלול

של מסלול כזה (זהו מקרה פרטי של הטענה על תתי־מסלולים קלים ביותר מהפרק המתאים), ולכן $v_{i-1} \in V_{i-1}$ (כי יש לו את המרחק המתאים מ־s).

החלק השני של ההוכחה נובע מהראשון: אם $s, u_1, \ldots, u_i$ מסלול כל שהוא בגרף השכבות, ניתן עכשיו להראות באינדוקציה שמתקיים $u_j \in V_j$ לכל $j \leq i$ (כאשר מגדירים $u_0 = s$). ■

בהינתן G ו־s, ניתן למצוא את גרף השכבות באופן הבא.

אלגוריתם למציאת גרף שכבות

קלט: גרף (מכוון) $G = (V, E)$ וצומת התחלה $s \in V$

פלט: גרף השכבות המתאים $G' = (V, E')$, וחלוקת הצמתים המושגים לשכבות $V_1, \ldots, V_k$

- **מבצעים** ב־G חיפוש לרוחב (BFS) מ־s, ושומרים את פונקצית המרחקים d שהחיפוש נותן
- מגדירים את E' להיות קבוצת כל הקשתות $uv \in E$ עבורן $d(v) = d(u) + 1$
- **לכל** i בין 0 ל־$|V| - 1$ מגדירים $V_i = \{v : d(v) = i\}$

נכונות האלגוריתם נובעת מהטענה האחרונה על קשתות ומסלולים. הסיבוכיות היא כשל החיפוש לרוחב, $O(|E|)$ (אנחנו עדיין מניחים שמתקיים $O(|V| + |E|) = O(|E|)$). עתה נגדיר את הזרימה שנרצה בגרף.

הגדרה: זרימה חוסמת שכבתית עבור $N = (G, s, t, c)$ היא זרימה חוקית g בגרף השכבות של G מ־s (ז"א זרימה ב־N שהיא 0 על הקשתות של G שאינן בגרף השכבות), שמקיימת בנוסף לכך שלכל מסלול קצר ביותר מ־s ל־t יש קשת e בתוכו עבורה $g(e) = c(e)$.

על מנת למצוא זרימה חוסמת בגרף שכבות, כל פעם ננסה לקחת מסלול מ־s. אם הוא מגיע ל־t אז נגדיל את הזרימה לפי המסלול (דבר שיאפשר לנו לחסר קשת מגרף השכבות בהמשך), ואם המסלול "נתקע" בצומת אחר אז נוכל "ללמוד" את הדבר ולחסר את הקשת שהובילה למבוי הסתום. באופן פורמלי נשתמש באלגוריתם הבא.

אלגוריתם למציאת זרימה חוסמת בגרף שכבות

קלט: רשת זרימה $N = (G, s, t, c)$ וגרף שכבות מתאים (V, E')

פלט: זרימה חוסמת שכבתית g

- **אתחול:** $g : E \rightarrow \mathbb{R}^+$ מאותחלת להיות פונקצית ה־0, ומציבים $F \leftarrow E'$
- **כל עוד** יש ב־F קשתות יוצאות מ־s

- **מבצעים** חיפוש לעומק מ־s ב־(V,F) רק עד שמגיעים ל־t או לצומת u שהוא עלה בעץ החיפוש
- יהיו $\langle v_0,\dots,v_k\rangle$ הצמתים במסלול שעליו החיפוש עבר
- **אם** $v_k = t$
  * יהי δ המינימום של "הקיבול הנותר" במסלול $\delta \leftarrow \min_{0<i\le k}(c(v_{i-1}v_i) - g(v_{i-1}v_i))$
  * **לכל** $0 < i \le k$ מציבים $g(v_{i-1}v_i) \leftarrow g(v_{i-1}v_i) + \delta$
  * **לכל** $0 < i \le k$ **עבורו** $g(v_{i-1}v_i) = c(v_{i-1}v_i)$ מסירים את $v_{i-1}v_i$ מ־F (קשת רוויה)
- **אחרת** ($v_k = u \neq t$)
  * מסירים את $v_{k-1}v_k$ מ־F (קשת מבוי־סתום)

למה (שמורות האלגוריתם): לאחר כל איטרציה של הלולאה המרכזית, g היא זרימה חוקית, $c(e) - g(e)$ הוא ערך הקיבול השיורי על קשתות $e \in E'$ (זה לא בהכרח נכון לקשתות אחרות של הגרף השיורי), ואין בגרף השכבות מסלול שיפור עבור g שעובר דרך קשתות שאינן ב־F (גם כאן זה לא בהכרח נכון למסלולים שמשתמשים בקשתות שאינן בגרף השכבות).

הוכחה: ראשית נזכיר שגרף השכבות הוא חסר מעגלים מכוונים, ולכן בכל רגע נתון גם תת־הגרף שלו (V,F) הוא כזה. בפרט אין ב־E' קשתות אנטי־מקביליות, ולכן $c(e) - g(e)$ הוא ערך הקיבול השיורי לזרימה g שמתאפסת מחוץ ל־E'. כמו כן החיפוש לעומק תמיד יעצור ב־t או בצומת ללא קשתות יוצאות של F (אין קשתות לאחור בגלל חוסר מעגלים, ומהעלה הראשון בעץ חיפוש לעומק לא יכולות להיות קשתות חוצות).

קל לוודא שתנאי השמורה מתקיימים בתחילת ההרצה (כאשר g זרימת האפס ו־(V,F) הוא גרף השכבות). בכל שלב ש־g מתעדכנת, g תהיה התוצאה של חיבור הזרימה שהיתה קודם עם מסלול שיפור, ולכן תישאר זרימה חוקית.

נותר להוכיח שאין מסלולי שיפור פרט לאלו שמשתמשים רק בקשתות ב־F. לשם כך בודקים את שני המקרים שבהם מסירים קשתות. המקרה של קשת רוויה מתבצע (לפי השמורות) כאשר בקשת של E' הזרימה שווה לקיבול הקשת, ז"א שזו מפסיקה להיות קשת בגרף השיורי (מכיוון שאין את הקשת ההופכית ב־E' וערכי g לא קטנים במהלך ריצת האלגוריתם, הקשת הזו גם לא תוכל לחזור).

כאשר מסירים קשת מבוי־סתום $v_{k-1}v_k$, הדבר אומר שאין קשתות יוצאות ב־F מ־v_k, ולכן v_k אינו יכול להיות על מסלול שיפור (כי עד עכשיו כל מסלולי השיפור היו חייבים להכיל רק קשתות מ־F). אבל מכך נובע שגם הקשת $v_{k-1}v_k$ אינה יכולה להיות על מסלול שיפור. ∎

למה (נכונות וסיבוכיות): האלגוריתם יעצור בזמן $O(|V||E|)$ ויחזיר זרימה חוסמת שכבתית.

הוכחה: כל איטרציה של הלולאה המרכזית לוקחת זמן $O(|V|)$: זהו הזמן עבור החיפוש לעומק, כי הוא אינו יכול לגלות יותר מ־$|V|$ צמתים עד שהוא מגיע לעלה הראשון של עץ החיפוש, ואין עבורו קשתות לאחור בגלל שהוא מתבצע על גרף השכבות (גרף השכבות הוא חסר מעגלים כמסקנה מיידית של הטענה על קשתות ומסלולים).

מכיוון שבכל איטרציה של הלולאה המרכזית לפחות קשת אחת מוסרת מ־F, יש לכל היותר $|E|$ איטרציות, ובסה"כ זמן ריצה של $O(|V||E|)$. הזמן של שאר פעולות האלגוריתם נבלע בזמן זה.

באשר לנכונות האלגוריתם, אם g לא היתה זרימה חוסמת, אז היה מסלול קצר ביותר מ־s ל־t (ומסלול כזה משתמש רק בקשתות של גרף השכבות) שבו אין אף קשת עם זרימה מקסימלית. אבל לפי הלמה על שמורות האלגוריתם דבר זה אומר בהכרח שהמסלול הנ"ל משתמש רק בקשתות של F, ולכן האלגוריתם לא היה עוצר (האלגוריתם עוצר רק כשאין כלל קשתות יוצאות מ־s). ■

עתה ננסח את האלגוריתם של דיניץ למציאת זרימה מקסימלית.

אלגוריתם דיניץ למציאת זרימה מקסימלית

קלט: רשת זרימה $N = (G, s, t, c)$ מעל הגרף המכוון $G = (V, E)$

פלט: זרימת מקסימום $f : E \to \mathbb{R}^+$

- **אתחול:** $f : E \to \mathbb{R}^+$ מאותחלת להיות פונקצית ה־0
- **כל עוד** אפשר לשפר את f ע"י הפרוצדורה הבאה
  - תהי $N_f = (G_f, s, t, c_f)$ הרשת השיורית של G ביחס ל־f
  - **מבצעים** את האלגוריתם למציאת גרף השכבות (V, E'_f) עבור G_f מ־s
  - **מבצעים** את האלגוריתם למציאת זרימה חוסמת g בגרף השכבות
  - **אם** g לא זרימת ה־0 **אז** מחליפים את f בחיבור שלה עם g לפי ההגדרה של חיבור זרימות, **ואחרת מסיימים את הלולאה**

טענה מיידית (נכונות בעצירה): האלגוריתם עוצר רק כאשר אין יותר מסלולים משפרים (כי אחרת הקצר ביותר מביניהם היה נמצא כולו בגרף השכבות ומביא לזרימה חוסמת גדולה מ־0), ולפי המשפט על זרימה מקסימלית וחתך מינימלי, זה קורה רק כאשר f היא זרימת מקסימלית. ■

השאלה שנותרה היא כמה איטרציות של הלולאה המרכזית צריך לכל היותר עד שהאלגוריתם עוצר. כמו בניתוח של אלגוריתם אדמונדס־קארפ, נסמן ב־$d_f(u)$ את המרחק (מספר קשתות במסלול קצר ביותר) מ־s ל־u. בכל הטענות הבאות נניח ש־g היא זרימה חוסמת בגרף השכבות של G_f מ־s, וש־h היא תוצאת השיפור של f לפי g. הטענות וההוכחות שלהן מזכירות את הטענות בדרך להוכחת הסיבוכיות של אלגוריתם אדמונדס־קארפ.

טענה (הפרש מרחקים דרך קשת): אם $uv \in E_h$, אז $d_f(v) - d_f(u) \leq 1$.

הוכחה: אם מתקיים גם $uv \in E_f$ אז הדבר נובע מאי־שוויון המשולש. אם לעומת זאת $uv \notin E_f$, אז הדבר אומר שהקשת נוספה כתוצאה מהגדלה של הזרימה בקשת $vu \in E_f$, ובפרט קשת זו נמצאת בגרף השכבות של G_f מ־s. אבל אז מתקיים לפי הטענה על קשתות ומסלולים בגרף השכבות ש־$d_f(v) - d_f(u) = -1$, שזה קטן מ־1. ■

טענה (גדילת המרחק): מתקיים $d_h(t) > d_f(t)$.

הוכחה: יהי $u_0, \ldots u_k$ כאשר $u_0 = s$ ו־$u_k = t$ מסלול קצר ביותר ב־G_h מ־s ל־t. לפי הטענה הקודמת מתקיים לכל $0 < i \leq k$ אי־השוויון $d_f(u_i) - d_f(u_{i-1}) \leq 1$. על כן לא ייתכן שמתקיים $d_f(t) > k$, והמקרה היחידי בו יכול להיות שוויון הוא כאשר לכל i מתקיים $d_f(u_i) = i$.

אבל, מכיוון שהנחנו ש־g היא זרימה חוסמת, לא יכול להיות שכל הקשתות מהצורה $u_{i-1}u_i$ הן קשתות גם ב־E_f וגם ב־E_h, בגלל שביצענו שיפור לפי זרימה חוסמת (ז"א שבכל מסלול קצר ביותר ב־G_f לפחות קשת אחת הוצאה במעבר ל־G_h). על כן, עבור $0 < j \leq k$ מתאים, הזוג $u_{j-1}u_j$ הוא קשת ב־E_h ולא ב־E_f. אבל זה יכול לקרוא רק אם ב־g היתה זרימה חיובית ב־u_ju_{j-1}, דבר שאינו ייתכן כי g אינה חיובית מחוץ לגרף השכבות (שלפי הטענה על קשתות ומסלולים אינו יכול לכלול קשת מהשכבה ה־j לשכבה ה־$j-1$).

על כן גם המקרה $d_f(t) = k$ אינו ייתכן, ולכן $d_h(t) = k > d_f(t)$. ■

מכאן אפשר לסכם.

מסקנה מיידית (סיבוכיות אלגוריתם דיניץ): מספר האיטרציות של הלולאה המרכזית באלגוריתם דיניץ חסום ע"י $|V|$ (כי $d_f(t)$ גדל כל איטרציה וחסום ע"י $|V|$), ולכן סיבוכיות האלגוריתם היא $O(|V|^2|E|)$, כי כל איטרציה לוקחת זמן $O(|V||E|)$ עבור מציאת הזרימה החוסמת בגרף השכבות (הסיבוכיות של פעולות אחרות, כגון מציאת גרף השכבות עצמו, נבלעת בביטוי זה). ■