# Solutions to Exercise 1

## Playing with vectors

Let us define $\alpha_i = p_1 - \frac{1}{n}$ for every $1 \le i \le n$. Then we know that $\sum_{i=1}^{n} \alpha_i = 0$ because $P$ is a probability distribution, and that $\sum_{i=1}^{n} |\alpha_i| \ge 2\epsilon$ by the assumption of the variation distance of $P$ from the uniform distribution.

Now we can use the Cauchy Schwartz inequality:

$$\sum_{i=1}^{n} p_i^2 = \sum_{i=1}^{n}(\frac{1}{n}+\alpha_i)^2 = \frac{1}{n} + \frac{2}{n}\sum_{i=1}^{n}\alpha_i + \sum_{i=1}^{n}\alpha_i^2 = \frac{1}{n} + \sum_{i=1}^{n}\alpha_i^2 \ge \frac{1}{n} + \frac{1}{n}(\sum_{i=1}^{n}|\alpha_i|)^2 \ge (1+4\epsilon^2)\frac{1}{n}$$

## Trying to ascertain uniformity

Let $Q$ be the uniform distribution over $\{1,\dots,n\}$, and $R$ be the uniform distribution over $\{1,\dots,\lceil\frac{1}{2}n\rceil\}$. It is not hard to see that $R$ is $\frac{1}{4}$-far from $Q$ for $n > 3$. So a good algorithm would accept $Q$ with probability at least $\frac{2}{3}$ and accept $R$ with probability at most $\frac{1}{3}$.

Let $\alpha$ be the probability that the algorithm accepts when all its sampled values came out different from each other. When the algorithm is provided with samples from $Q$, the probability that this does not happen (by a simple union bound) is at most $\binom{\sqrt{n}/10}{2} \cdot \frac{1}{n} < \frac{1}{100}$. Therefore the acceptance probability for $Q$ is at most $\frac{99}{100}\alpha + \frac{1}{100}$.

When the algorithm is provided with samples from $R$, the probability for not all values being different is at most $\binom{\sqrt{n}/10}{2} \cdot \frac{2}{n} < \frac{1}{100}$, and so the acceptance probability here is at least $\frac{99}{100}\alpha$. There exists no $\alpha$ for which both $\frac{99}{100}\alpha + \frac{1}{100} \ge \frac{2}{3}$ and $\frac{99}{100}\alpha \le \frac{1}{3}$.

## Quicksort exposed

Let $s_1,\dots,s_n$ and $X_{i,j}$ be as in the hint given for the question. The event $X_{i,j} = 1$ (i.e. the event that the $i$'th smallest item was compared with the $j$'th smallest item) occurs if and only if either $s_i$ or $s_j$ was the first item selected as pivot among $s_i,\dots,s_j$. The probability for that is exactly $\frac{2}{j-i+1}$ (as long as none of $s_i,\dots,s_j$ was selected as pivot, the entire range will continue to be contained in a single segment on which the algorithm recurses).

As the expectation of the total number of comparisons is the sum of expectations of the indicator variables $\sum_{1\le i<j\le n} \mathrm{E}[X_{i,j}]$, the rest is arithmetic (we use here $\sum_{k=1}^{m}\frac{1}{k} = \ln(m) + O(1)$, provable e.g. by approximating with an integral):

$$\begin{aligned}\sum_{1\le i<j\le n} \mathrm{E}[X_{i,j}] &= \sum_{1\le i<j\le n}\frac{2}{j-i+1} = 2\sum_{i=1}^{n}\sum_{k=1}^{n-i}\frac{1}{k+1} \\ &= 2\sum_{k=1}^{n-1}\frac{n-k}{k+1} = 2(1-n+(n+1)\sum_{k=1}^{n-1}\frac{1}{k+1}) = 2n(\ln(n)+O(1)).\end{aligned}$$

## Troubles moving forward

$t_k$ was defined as the expected difference between the first $i$ for which $X_i = k-1$ occurred and the first $j$ for which $X_j = k$ occurred. However, because of the way the $X_i$ were defined (where the probabilities for $X_{i+1}$ depend only on $X_i$ and not on any $X_k$ for $k \leq i$), $t_k$ gives also the expected distance between the $r$'th $i$ for which $X_i = k-1$ occurred (for any $r$) and the smallest $j > i$ for which $X_j = k$ occurred.

Now for $k > 1$ assume that we calculated $t_1, \ldots, t_{k-1}$ (remember that $t_1 = 3$ was calculated in the question), and set $i$ to be the smallest index for which the event $X_i = k-1$ occurred. We shall analyse the distribution for $j$, the smallest index for which $X_j = k$ occurred, conditioned on the value of $i$.

Denote by $i_r$ the $r$'th index for which $X_{i_r} = k-1$, where $i_1 = i$. With probability exactly $\frac{1}{3}$ we have $X_{i+1} = k$ and hence $j = i+1$. With the remaining probability $\frac{2}{3}$ we have $X_{i+1} = k-2$ (remember that $k > 1$). In this case the expectation of $i_2$ is $i + 1 + t_{k-1}$, and again with probability $\frac{1}{3}$ we have $j = i_2 + 1$ and with probability $\frac{2}{3}$ we have $X_{i_2+1} = k-2$. Continuing this argument, we can calculate $t_k$ in terms of $t_{k-1}$:

$$
\begin{aligned}
t_k &= \sum_{r=1}^{\infty} \mathrm{E}[j | i_r < j < i_{r+1}] \cdot \Pr[i_r < j < i_{r+1}] \\
&= \sum_{r=1}^{\infty} (1 + (r-1)(t_{k-1}+1)) \cdot \frac{1}{3}(\frac{2}{3})^{r-1} \\
&= \frac{-t_{k-1}}{3} \sum_{r=1}^{\infty} (\frac{2}{3})^{r-1} + \frac{t_{k-1}+1}{3} \sum_{r=1}^{\infty} (\frac{2}{3})^{r-1} r = 2t_{k-1} + 3.
\end{aligned}
$$

Solving this recurrence gives us $t_k = 3 \cdot 2^k - 3$.

## Solutions to Exercise 2

### Three-sum-free

Let $p$ be a prime number that is large enough to satisfy $(\frac{1}{4}-\frac{8}{p})|A| > \lceil\frac{1}{4}|A|\rceil - 1$ (and in particular $p > |A|$). Let us denote $p = 8k + l$ where $0 < l < 8$. We first note that in modulo $p$ arithmetic, the set $C = \{k, k+1, \ldots, 3k-1\}$ has no three members whose sum is equal to another member (the sum of any three members in this range is either at least $3k$ modulo $p$ or at most $3(3k-1) - p < k$). Now define for $1 \le i < p$ by $iC \pmod p$ the set of multiples by $i$, $\{ic \pmod p | c \in C\}$ (we always use the representative numbers between $0$ and $p-1$). We have that $|iC| = |C| = 2k$ and also that $iC$ has no three members whose sum is in $iC$ (whether we use molulo $p$ arithmetic or regular integer arithmetic).

Now we choose $0 < i < p$ uniformly at random and set $B = (iC) \cap A$. The resulting set clearly has the "three-sum-free" property, and it remains to bound the expectation of its size. For every $a \in A$, the probability for $a \in B$ is $\frac{2k}{p-1}$ (by the same arguments as in the proof for sum-free sets in the lecture notes; there are exactly $2k$ choices for $i$ such that $a/i \in C \pmod p$). Therefore by the linearity of expectation the expected size of $B$ is $\frac{2k}{p-1}|A| \ge (\frac{1}{4}-\frac{8}{p})|A| > \lceil\frac{1}{4}|A|\rceil - 1$. Therefore there is a choice of $i$ for which $|B|$ is greater than $\lceil\frac{1}{4}|A|\rceil - 1$, and hence at least $\frac{1}{4}|A|$.

### Ascertaining uniformity

Let $X_{i,j}$ be the indicator variable for $a_i = a_j$, and let $X = \sum_{1\le i<j\le k} X_{i,j}$ be the random variable for the number of pairs for which $a_i = a_j$. The probability for $X_{i,j} = 1$ is clearly $\sum_{l=1}^{n} p_l^2 = \|P\|$, and so $\mathrm{E}[X] = \|P\|\binom{k}{2}$ (indeed we happen to use here a somewhat unconventional notation for $\|P\|$). Now let us bound $\mathrm{V}[X]$. If $\{i_1, j_1\}$ and $\{i_2, j_2\}$ are disjoint (as sets, not intervals) then $X_{i_1,j_1}$ and $X_{i_2,j_2}$ are independent. If these two sets share exactly one member, then $\mathrm{Cov}[X_{i_1,j_1}, X_{i_2,j_2}] \le \mathrm{E}[X_{i_1,j_1}X_{i_2,j_2}] = \sum_{l=1}^{n} p_l^3 \le \|P\| \max_{1\le l\le n} p_l \le \|P\|^{3/2}$. Also, it is not hard to see that $\mathrm{Cov}[X_{i,j}, X_{i,j}] \le \|P\|$. Using all this we obtain:

$$\mathrm{V}[X] = \sum_{i_1<j_1, i_2<j_2} \mathrm{Cov}[X_{i_1,j_1}, X_{i_2,j_2}] \le \binom{k}{2}\|P\| + 6\binom{k}{3}\|P\|^{3/2} < \binom{k}{2}\|P\| + k^3\|P\|^{3/2}.$$

Using Chebyshev's inequality, this means that

$$\Pr\big[|X - \binom{k}{2}\|P\|| > \epsilon^2\|P\|\binom{k}{2}\big] \le (\binom{k}{2}\|P\| + k^3\|P\|^{3/2})/\|P\|^2\epsilon^4\binom{k}{2}^2 < \frac{1}{\|P\|100n} + \frac{1}{\|P\|^{1/2}10\sqrt{n}}.$$

Now we analyse two cases. The first case is where $P$ is uniform. In this case $\|P\| = \frac{1}{n}$, the inequality above shows that in particular we will have $X \le (1+\epsilon^2)\binom{k}{2}$ with probability more than $\frac{2}{3}$, which will cause the algorithm to say "yes" with at least this probability.

The second case is where $P$ is $\epsilon$-far from uniform. Because $\|P\| \ge \frac{1}{n}$ always, in this case with probability more than $\frac{2}{3}$ we will have $X \ge \binom{k}{2}\|P\| - \epsilon^2\|P\|\binom{k}{2}$. The question "Playing with vectors" from Exercise 1 tells us that here in fact $\|P\| > (1+4\epsilon^2)\frac{1}{n}$, and so the lower bound on $X$ would cause the algorithm to say "no" with at least this probability.

## Solutions to Exercise 3

### Editing strings

We assume that $n > N$ for $N = \max\{N_1, N_2\}$ that will be chosen later (depending on $k$ and $\epsilon$). Assume also that $\epsilon$ is such that $1/\epsilon$ is an integer larger than 2 (otherwise decrease $\epsilon$ accordingly), set $l = 8k^2/\epsilon - 1$ and $t = \lfloor n/l \rfloor$, and delete the last $n - tl$ bits of $v$. Choosing $N_1 = 8l/\epsilon$ makes sure that no more than $\frac{1}{8}\epsilon n$ bits were deleted.

Next, we look at each substring $v_{it+1}v_{it+2}\dots v_{it+l}$ for $0 \le i < t$. We uniformly and independently choose $0 \le j_i < k$, and we delete the first $j_i$ bits and the last $k - 1 - j_i$ bits of this substring (note that the reminder is a consecutive substring whose size is a multiple of $k$). All in all we have deleted in this stage $tk$ bits, making the total number of deleted bits less than $\epsilon n$. We call the resulting string $v'$, and show that with positive probability it will be the string we need.

For a fixed $w \in \{0,1\}^k$ let us analyse $\mathcal{T}_{v'}(w)$. Its expectancy is the fraction of the copies of $w$ among the substrings of type $v_{i+1}\dots v_{i+k}$, where $i$ is any number between 0 and $tl - 1$ whose residue modulo $l$ is at most $l - k$. By comparison, $\mathcal{S}_v(w)$ is the fraction of the copies of $w$ among the substrings of type $v_{i+1}\dots v_{i+k}$, where $i$ is any number between 0 and $n - k$ (with any residue). The above choice of $l$ and $N_1$ makes sure that $|\mathcal{S}_v(w) - \mathrm{E}[\mathcal{T}_{v'}(w)]| \le \frac{\epsilon}{2}$, because less than a $\frac{1}{3}\epsilon n$ of the possible $i$ counted in $\mathcal{S}_v(w)$ are not counted in $\mathrm{E}[\mathcal{T}_{v'}(w)]$.

Now we choose $N_2$ so that with high probability $\mathcal{T}_{v'}(w)$ will be close to its expectation. Note that $X = t\mathcal{T}_{v'}(w)$ is actually the sum of $t$ independent random variables variables $X_1, \dots, X_t$, where $X_i = \mathcal{T}_{v_{li-l+j_i+1}\cdots v_{li-k+j_i}}(w)$. In particular $|X_i| \le 1$ always, and we can choose $N_2$ so that $t$ would be large enough to ensure by a large deviation inequality that $\Pr[|X - \mathrm{E}[X]| > \frac{\epsilon}{2}t] < 2^{-k}$. Using the Chebyshev inequality would work here, but for a smaller $N_2$ it is better to use the inequality resulting from the martingale exposing $X_1, \dots, X_t$.

Using now the union bound for all possible $2^k$ choices of $w$ we get that with positive probability $|\mathcal{T}_{v'}(w) - \mathrm{E}[\mathcal{T}_{v'}(w)]| \le \frac{\epsilon}{2}$ for all $w$, and we are done.

### Satisfying all

We choose independently the value of every variable $x_i$ from $\{0,1\}$. However, instead of uniformly, we make $x_i = 0$ with probability 0.3 and $x_i = 1$ with probability 0.7. Now, every clause with between two and six literals has only positive literals, and hence its probability to not be satisfied is bounded by $(0.3)^2 < \frac{1}{4e}$. A clause with more the six literals may (at the worse case) have only negative literals, and so its probability to not be satisfied is bounded by $(0.7)^7 < \frac{1}{4e}$.

For every clause $C$ we define the event $E_C$ as that of $C$ not being satisfied. This event is independent of all events $E_{C'}$ concerning clauses $C'$ that do not share any of the variables of $C$, and (by the question statement) these include all but at most three other events. This together with the bound $\Pr[E_C] < \frac{1}{4e}$ for every $C$ allows us to use the symmetric version of the local lemma, and conclude that with positive probability none of the above events takes place, and hence all clauses are satisfied.

## Solutions to Exercise 4

### Playing with FKG

Let $\alpha$ be the maximum value of $h(A)$, and define $g$ by $g(A) = \alpha - h(A)$. Now $g$ is non-negative and monotone non-decreasing, and we can use the original FKG theorem to conclude

$$\Big(\sum_{A\subseteq S}\mu(A)f(A)\Big)\Big(\sum_{A\subseteq S}\mu(A)g(A)\Big) \le \Big(\sum_{A\subseteq S}\mu(A)f(A)g(A)\Big)\Big(\sum_{A\subseteq S}\mu(A)\Big).$$

Substituting for $g$ and opening parentheses we get

$$\Big(\sum_{A\subseteq S}\mu(A)f(A)\Big)\Big(\sum_{A\subseteq S}\mu(A)\alpha\Big) - \Big(\sum_{A\subseteq S}\mu(A)f(A)\Big)\Big(\sum_{A\subseteq S}\mu(A)h(A)\Big) \le$$

$$\le \Big(\sum_{A\subseteq S}\mu(A)f(A)\alpha\Big)\Big(\sum_{A\subseteq S}\mu(A)\Big) - \Big(\sum_{A\subseteq S}\mu(A)f(A)h(A)\Big)\Big(\sum_{A\subseteq S}\mu(A)\Big).$$

The rest is a simple manipulation and remembering that $\alpha$ is a constant.

### Just wandering

For ease of notation we assume that $u$ not a neighbor of $t$ (by our assumptions $u$ and $v$ cannot both be neighbors of any vertex, and we can "swap" their roles by moving from $\sigma$ to its inverse), and note also that $u$ and $v$ cannot be equal to $t$ by our assumptions on $\sigma$. Let $\phi : V \to \mathbb{R}$ be the harmonic function with boundary $\{s,t\}$ such that $\phi(s) = 1$ and $\phi(t) = 0$. Now $\phi_\sigma$ defined by $\phi_\sigma(x) = \phi(\sigma(x))$ is also harmonic with the same boundary conditions as $\phi$ (this is easy to check), and so $\phi_\sigma = \phi$ and in particular $\phi(u) = \phi(\sigma(u)) = \phi(v)$.

Now we define the graph $G'$ that is obtained by "fusing" $u$ and $v$. Formally we define $V' = V \setminus \{u\}$, and $E'$ to contain all edges $\{xy \in E | x, y \neq u\}$ plus the edges $\{vy | uy \in E\}$. We also define $\phi' = \phi|_{V'}$. We claim that $\phi'$ is harmonic for $G'$ with boundary $\{s,t\}$ such that $\phi(s) = 1$ and $\phi(t) = 0$: All conditions relating to it being harmonic follow immediately (remember that $\phi(v) = \phi(u)$), apart from the condition $\phi'(v) = \frac{1}{d_{G'}(v)} \sum_{vx\in E'} \phi'(x)$.

For the last one we write:

$$\begin{aligned}\frac{1}{d_{G'}(v)}\sum_{ux\in E'}\phi'(x) &= \frac{1}{d_G(u)+d_G(v)}\Big(\sum_{ux\in E}\phi(x) + \sum_{vx\in E}\phi(x)\Big) \\ &= \frac{1}{d_G(u)+d_G(v)}(d_G(u)\phi(u) + d_G(v)\phi(v)) = \phi(v) = \phi'(v),\end{aligned}$$

and we are done.

Since clearly $|E| = |E'|$ (we used implicitly everywhere that $u$ and $v$ are not adjacent and share no neighbors), and $R_{st}$ is calculated by the average over the neighbors of $t$ of the harmonic function $\phi$ which is 1 on $s$ and 0 on $t$ (see the material taught in class), we have that both $G$ and $G'$ have the same $R_{st}$ (as $\phi(x) = \phi'(x)$ on the relevant vertices) and hence the same commute time $k_{st}$.

### Independent in triples

Let $Y_1, \ldots, Y_{k+1}$ be random variables representing $k+1$ uniform and independent random bits. Now let $V \subset \{0,1\}^{k+1}$ be the set of all binary vectors with an odd number of 1's. It is easy to see that $|V| = 2^k$. Now for every vector $v = (a_1, \ldots, a_k) \in V$ we define the random variable $X_v = \sum_{i=1}^{k+1} a_i Y_i$, and claim that these are independent in triples.

The thing to note is that for every $u, v \in V$ the number of 1's in the bitwise XOR of these vectors is even, and so there is no $w \in V$ such that $u \oplus v = w$. Therefor, for every three distinct $u, v, w \in V$ we have that the probability for $X_u \oplus X_v \oplus X_w = 1$ is exactly $\frac{1}{2}$. Also each of $X_u$, $X_v$ and $X_w$ gets 1 with probability $\frac{1}{2}$, and each XOR of two of these three variables gets 1 with probability $\frac{1}{2}$. The above can all happen only if the probability for every possible outcome of the triple $(X_u, X_v, X_w)$ is $\frac{1}{8}$, implying that the three variables are independent.