Algorithmic Combinatorics — WS 2014 / 2015

Tibor Szabó
Shagnik Das

# Solutions to Exercise Sheet 14

**Exercise 1.** Suppose one can compute the determinant of an $n \times n$ matrix with $O(n^\omega)$ arithmetic operations. There is a probabilistic algorithm that tests for the *existence* of perfect matchings in an $n$-vertex graph with $O(n^\omega)$ operations.

Using this algorithm, develop an algorithm for *finding* a perfect matching in an $n$-vertex graph with $m$ edges that requires $O(mn^\omega \log n)$ operations.

Solution: The first step to finding a perfect matching is determining whether one exists. We may therefore first run the randomised algorithm and, if it returns the existence of a perfect matching, try to find it. To remove any possible ambiguity, we will suppose that both the vertices and edges of the graph are ordered (arbitrarily, perhaps), so that we may talk of the *first* vertex (or edge) in some set of vertices (or edges).

To find the perfect matching, we start with an empty matching and proceed edge-by-edge, trying to find an edge we can enlarge our current matching by. Of course, we need to have a perfect matching at the end of the process, so we must ensure that our current matching can always extend to a perfect matching. Luckily for us, we have a probabilistic algorithm that can determine this for us.

Thus we start by setting the initial matching to be empty, $M_0 = \emptyset$. Given $1 \leq i \leq \frac{n}{2}$ and a partial matching $M_{i-1}$ that can be extended to a perfect matching, let $v_i$ be the first vertex not spanned by $M_{i-1}$. We now search for an edge incident to $v_i$ that we can add to the matching $M_{i-1}$ (such an edge must exist, as we know $M_{i-1}$ extends to a perfect matching).

Consider the edges between $v_i$ and its neighbours not spanned by $M_{i-1}$, and consider each such edge $e$ in order. Let $M_i' = M_{i-1} \cup \{e\}$, and $G' = G \setminus M_i'$. Run the randomised algorithm to see if $G'$ has a perfect matching. If not, then we discard $e$ and try the next available edge. If it does, though, this means that $M_i'$ extends to a perfect matching, so we set $M_i = M_i'$ and iterate with the next available vertex $v_{i+1}$.

By the end of this algorithm, we will have found a perfect matching $M_{\frac{n}{2}}$, provided we always got the correct answer when asking if a matching exists in the subgraphs $G \setminus M_i'$. Each individual application of the randomised algorithm is correct with probability $\frac{1}{2}$, but we have to call upon this algorithm repeatedly, and will with high probability get a false result at some point. Hence, in order to ensure our algorithm works with positive probability, we can repeatedly run the randomised algorithm to boost its accuracy.

By repeating the individual randomised test $k$ times, the error probability is reduced to $2^{-k}$. Note that we need to test the existence of a matching in at most $m$ subgraphs, since after an edge is discarded it is never considered again. Thus, by repeating each individual randomised test at most $k$ times (if the test tells us a matching exists, it cannot be an error, hence we can add the edge in question to the matching and proceed without any further repetitions of that test), the expected number of overall errors is at most $m2^{-k}$. For $k = \lceil \log_2 m \rceil + 1$, this expectation is at most $\frac{1}{2}$. Hence, by Markov's Inequality, the probability of the algorithm failing is at most $\frac{1}{2}$.

Finally, this procedure calls the randomised existence algorithm at most $mk$ times, with each call requiring $O(n^\omega)$ operations. Since $k = O(\log m)$ and $m = O(n^2)$, it follows that the above algorithm requires $O(mn^\omega \log n)$ operations.

**Exercise 2.** The goal of this exercise is to prove $m(k) = O(k^2 2^k)$ via a randomised construction. Fix the ground set of elements $[2n]$, and consider the random hypergraph $\mathcal{F}$ with $m$ edges, $F_1, F_2, \ldots, F_m$, chosen independently and uniformly at random (with repetition) from the family $\binom{[2n]}{k}$ of all $k$-sets in $[2n]$.

(i) Let $\chi$ be a (fixed) red/blue colouring of the elements $[2n]$. Show that, for each $i$, the probability of $F_i$ being monochromatic is at least $2\binom{n}{k}/\binom{2n}{k}$.

(ii) If $p$ is the lower bound from (i), show that $p = 2\prod_{j=0}^{k-1} \frac{n-j}{2n-j} \geq \left(\frac{n-k}{2n-k}\right)^k$. Given $1 - x \leq e^{-x} \leq 1 - \frac{1}{2}x$ for $x$ sufficiently small, show that $p \geq 2^{-k}e^{-2k^2/(2n-k)}$ if $n$ is sufficiently large with respect to $k$.

(iii) Deduce that the probability of there being a proper colouring of $\mathcal{F}$ is at most $2^{2n}(1-p)^m$.

(iv) By choosing appropriate values for $m$ and $n$ in terms of $k$, show that there exists a non-two-colourable $k$-graph $\mathcal{F}$ with $O(k^2 2^k)$ sets.

Solution:

(i) Suppose $\chi$ has $r$ red elements, and $2n - r$ blue elements. There are then $\binom{r}{k}$ all-red $k$-sets and $\binom{2n-r}{k}$ all-blue $k$-sets out of the total $\binom{2n}{k}$ $k$-sets. Since $F_i$ is chosen uniformly at random, the probability that $F_i$ is monochromatic under $\chi$ is

$$p := \frac{\binom{r}{k} + \binom{2n-r}{k}}{\binom{2n}{k}} \geq \frac{2\binom{n}{k}}{\binom{2n}{k}},$$

where the inequality follows from the convexity of $\binom{x}{k}$ for $x \geq k-1$.

(ii) Since $\binom{n}{k} = (k!)^{-1}\prod_{j=0}^{k-1}(n-j)$, we can expand the binomial coefficients to get

$$p \geq \frac{2\binom{n}{k}}{\binom{2n}{k}} = \frac{2(k!)^{-1}\prod_{j=0}^{k-1}(n-j)}{(k!)^{-1}\prod_{j=0}^{k-1}(2n-j)} \geq \prod_{j=0}^{k-1} \frac{n-j}{2n-j}.$$

These terms are decreasing in $j$, so $\frac{n-j}{2n-j} \geq \frac{n-k}{2n-k}$, and thus we have

$$p \geq \left(\frac{n-k}{2n-k}\right)^k = \left(\frac{1}{2} - \frac{k}{4n-2k}\right)^k = 2^{-k}\left(1 - \frac{k}{2n-k}\right)^k.$$

Provided $k = o(n)$, we have $1 - \frac{k}{2n-k} \geq e^{-2k/(2n-k)}$, and hence $p \geq 2^{-k}e^{-2k^2/(2n-k)}$.

(iii) A colouring $\chi$ is proper if and only if each of the edges $F_i$ is not monochromatic. Since the sets $F_i$ are chosen independently, these events are independent, and each occurs with probability at most $1-p$. Hence the probability that a given colouring $\chi$ is proper is at most $(1-p)^m$. Taking a union bound over all $2^{2n}$ possible colourings of the groundset $[2n]$, the probability of there existing a proper colouring is at most $2^{2n}(1-p)^m$.

(iv) We can further bound the probability of there being a proper colouring by

$$2^{2n}(1-p)^m \leq 2^{2n}e^{-pm}. \qquad (*)$$

If this bound is less than 1, then it follows that with positive probability we have a non-two-colourable $k$-uniform hypergraph with $m$ sets. This bound is less than 1 if and only if $e^{pm} > 2^{2n}$, or $m > 2np^{-1}\ln 2$. By our lower bound on $p$, it suffices to have $m > 2^{k+1}e^{2k^2/(2n-k)}n\ln 2$. At this point it is enough to note that if we take $n = 2k^2$, say, then we get the desired bound of $m = O(k^2 2^k)$. In the sequel, we try to justify a bit further why this is approximately the right choice for $n$.

To achieve a good bound on $m(k)$, we should try to make $m$ as small as possible. Since $k$ is fixed, we only need to minimise $e^{2k^2/(2n-k)}n$. Taking logs, this is equivalent to minimising $\frac{2k^2}{2n-k} + \ln n$. Introduce a change of variables by setting $n = \frac{k^2}{\omega} + \frac{k}{2}$, after which we seek to minimise

$$2\omega + \ln\left(\frac{k^2}{\omega} + \frac{k}{2}\right) = 2\omega - \ln\omega + 2\ln k + \ln\left(1 + \frac{\omega}{2k}\right). \qquad (**)$$

Note that if we take $\omega = \frac{1}{2}$, $(**)$ becomes $2\ln k + 1 + \ln\left(1 + \frac{1}{4k}\right) \sim 2\ln k + 1 + \frac{1}{4k} \leq 2\ln k + 2$.

Since the expression in $(**)$ can be bounded below by $2\omega - \ln\omega \geq \omega$, we may therefore assume $0 < \omega \leq 2\ln k + 2$. In this range, the last summand in $(**)$ is asymptotically $\omega/(2k)$, and so we need to minimise $2\ln k + (2 + 1/(2k))\omega - \ln\omega$. Setting the derivative equal to zero shows that there is a minimum when $\omega = (2 + 1/(2k))^{-1} \approx \frac{1}{2}$, which corresponds to $n \approx 2k^2$.

**Exercise 3.** Improve either[1] the upper or lower bound on $m(k)$ to $k2^k$.[2]

Solution: If I had a solution, I'm afraid this wouldn't be where I'd write it down!

[1]Bonus points for doing both!

[2]A successful solution might constitute a good start to a Ph.D.

**Exercise 4.** Show that the bound in the Erdős–Selfridge theorem is tight, i.e. $\tilde{m}(k) = 2^{k-1}$.

[Hint: redisnoC a yranib eert fo htped $k-1$, dna dliub a elbatius hpargrepyh. esU eht yrtemmys fo eht eert ot dnif a ygetarts rof M.]

Solution: We will give a couple[3] of constructions of $k$-uniform hypergraphs $\mathcal{F}$ with $2^{k-1}$ edges on which Maker (M) has a winning strategy, thus showing $\tilde{m}(k) \leq 2^{k-1}$. Together with the lower bound from lecture, this shows $\tilde{m}(k) = 2^{k-1}$.

For the first, take the ground set to be the vertices $V$ of a rooted binary tree $T$ of depth $k-1$. That is, there is a root vertex $v_0$ with two children — we shall think of them as a left- and right-child of $v_0$. Each child will itself have two children, and so on until we reach level $k-1$, where we will have $2^{k-1}$ leaf vertices. The hyperedges of the hypergraph $\mathcal{F}$ will be the $2^{k-1}$ paths from the root vertex $v_0$ to a leaf of $T$.

Maker plays first, and should start by claiming the root. In fact, Maker's $i$th move will always be at depth $i-1$, and so Maker's vertices will grow as a path starting from the root. If Maker can proceed in this fashion for $k$ turns, she will have a path of length $k$, which must end at a root. Hence this will constitute a winning set, giving the desired winning strategy.

To see that Maker can do this, suppose that after her $i$th turn, Maker has successfully occupied a path of length $i$, starting at the root $v_0$ and ending at some vertex $v_{i-1}$ at depth $i-1$, such that Breaker (B) has not occupied any descendent of (i.e. vertex below) of $v_{i-1}$. This is certainly true for $i=1$, since Breaker has not even played at this point.

On his $i$th turn, Breaker can occupy a vertex in either the left- or right-subtree of $v_{i-1}$ (or neither), but as they are disjoint, he cannot occupy both. On her next turn, Maker can then choose either the right- or left-child of $v_{i-1}$ accordingly, thus extended her path and maintaining the property that Breaker has not occupied any vertex below the last vertex $v_i$. By doing this for $k$ turns, Maker wins the game.

A similar idea lies behind the second example. Let the ground set to be the disjoint union $X = S_0 \cup S_1 \cup S_2 \cup \ldots \cup S_{k-1}$, where $|S_0| = 1$ and $|S_i| = 2$ for $1 \leq i \leq k-1$. Take the hypergraph to be $\mathcal{F} = S_0 \times S_1 \times \ldots \times S_{k-1}$, and observe that $|\mathcal{F}| = 2^{k-1}$.

Maker starts by choosing the sole element of $S_0$. Then in each subsequent move, Breaker must choose some element from a set $S_i$, $1 \leq i \leq k-1$. Whenever Breaker does this, Maker responds by choosing the other element from the same set $S_i$. In this way, Maker guarantees that she occupies one element from each set $S_i$, which gives her some edge $F \in \mathcal{F}$, and thus she wins.

It may be interesting to note that in both of these examples, not only does Maker win, but she wins in $k$ moves, which is clearly as fast as possible. However, if you remove even one of the winning sets in $\mathcal{F}$, then Maker cannot win no matter how long they play!

[3] Only one is needed to solve the problem, but the more, the merrier, right?

**Exercise 5.** Suppose $M$ and $B$ play a more symmetric game on a $k$-graph $\mathcal{F}$. In every turn, first $M$ colours one element red, and then $B$ colours one element blue. The first player to have coloured (with his or her own colour) every element in some $F \in \mathcal{F}$ wins the game. If all the elements get coloured without either player managing to win, the game is a draw.

Show that for any $k$-graph, $M$ has a strategy that guarantees her either a win or a draw.

Solution: This is a finite game of perfect information, and so either one player has a winning strategy, or both players can guarantee a draw. Suppose for contradiction $B$ has a winning strategy. This means that, as second player, he can respond to any sequence of moves $M$ makes and guarantee that he wins the game at the end.

$M$ can now steal this strategy — she plays her first move $x_0$ arbitrarily, forgets about it, and then pretends she is the second player and uses $B$'s winning strategy. If at any point during the game the strategy requires her to claim $x_0$, then she can claim some other available element $x_1$ arbitrarily, since she has already claimed $x_0$. Proceeding in this fashion, she can respond to any sequence of moves $B$ makes, and then should win the game. However, if $B$ is using his winning strategy, then he should also win the game. As at most one player can win the game, it follows that $B$ cannot have a winning strategy.

(Intuitively, the reason this strategy stealing works is that $M$ can never be disadvantaged by having claimed an extra element. Thus this method cannot show that Black cannot have a winning strategy in chess.)

**Exercise 6.** Let $G$ be a bipartite graph with $n$ vertices. Show that if every vertex of $G$ is assigned a list of $\log_2(n)$ colours, $G$ has a proper list-colouring.

Solution: (We shall assume that $n \geq 3$, since the result is false for $n = 2$, but graphs on 2 vertices are not very interesting anyway.)

Let $G = (V_1 \cup V_2, E)$, and let $C = \cup_v L(v)$ be the set of all colours that appear in the lists. We randomly partition $C = C_1 \cup C_2$ by assigning each colour $c \in C$ to either $C_1$ or $C_2$ independently and uniformly at random.

Once we have this random partition, colour every vertex $v \in V_i$ with an arbitrary colour from $L(v) \cap C_i$. Provided we can choose a colour for each vertex (that is, that the intersection is not empty), this must give a proper list-colouring, since each edge will have one colour from $C_1$ and the other from the disjoint set $C_2$.

Given a vertex $v \in V_i$, define an indicator random variable

$$X_v = \begin{cases} 1 & \text{if } L(v) \cap C_i = \emptyset \\ 0 & \text{otherwise} \end{cases}.$$

$X_v$ indicates when $v$ cannot be coloured, so $X = \sum_v X_v$ counts how many uncolourable vertices there are. We have $\mathbb{P}(X_v = 1) = 2^{-|L(v)|} = 2^{-\log_2 n} = 1/n$, and so

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_v X_v\right] = \sum_v \mathbb{E}[X_v] = \sum_v \mathbb{P}(X_v = 1) = n \cdot \frac{1}{n} = 1.$$

However, since $n \geq 3$, we may without loss of generality assume $|V_1| \geq 2$. When we have $C_1 = \emptyset$ and $C_2 = C$, which occurs with positive probability ($2^{-|C|}$, to be precise), then the events $X_v$ for $v \in V_1$ all occur simultaneously, and hence $X = |V_1| \geq 2$. Since $X$ is a non-negative variable and has average $\mathbb{E}[X] = 1$, it follows that we must have $X = 0$ with positive probability as well.

Thus there exists a partition $C = C_1 \cup C_2$ such that for $i = 1, 2$ and for every $v \in V_i$, we can colour $v$ with some colour from $L(v) \cap C_i$, which gives rise to a proper list-colouring.