# Exercise Set VI, Algorithms II 2024

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

**1** In class we saw that Karger's min-cut algorithm implies that an undirected graph has at most $\binom{n}{2}$ minimum cuts. Show that this result is tight by giving a graph with $n$ vertices and $\binom{n}{2}$ minimum cuts.

**2** Change Karger's algorithm so that it also works for edge-weighted graphs. Also adapt the analysis to prove that it still returns any min cut $(S^*, \overline{S^*})$ with probability at least $1/\binom{n}{2}$. (Hence, edge-weighted graphs also have at most $\binom{n}{2}$ min cuts.)

**3** Consider an $n$-by-$n$ bipartite graph $G = (X \cup Y, E)$ and let $A$ be the adjacency matrix where we have $A_{ij} = \begin{cases} x_{ij} & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}$ (as in Lecture 13). In that lecture, we saw that we could replace each $x_{ij}$ by a random number and check whether $\det(A) \neq 0$ to see whether $G$ has a perfect matching. A beautiful alternative approach was introduced by a very influential paper by Mulmuley, Vazirani, and Vazirani'87 who considered *isolating weight functions*.

**3a** A weight function $w : E \to \mathbb{N}$ is *isolating* if the min-weight perfect matching is unique. Show that if $w$ is isolating then

$$\det(\hat{A}) \neq 0 \Leftrightarrow G \text{ has a perfect matching},$$

where $\hat{A}$ is the matrix obtained from $A$ by replacing each variable $x_e$ by $2^{w(e)}$.

**3b** *(*)* In this exercise, we prove the "Isolation Lemma". Show that if we form the weight function $w$ by letting, for each edge $e$ independently, $w(e)$ be a random number in $\{1, \dots, 2|E|\}$, then the probability that $w$ is isolating is at least $1/2$.

*(Hint: for each edge $e$, consider $\alpha_e = \min_{M \ni e} w(M \setminus \{e\})$ and $\beta_e = \min_{M \not\ni e} w(M)$. What is the probability that $\alpha_e + w(e) = \beta_e$?)*

**4** In this exercise, we are going to analyze a beautiful and simple randomized algorithm for the 2-SAT problem (due to Papadimitriou'91). Recall that in the 2-SAT problem we have $n$ variables $x_1, x_2, \ldots, x_n$ and $m$ clauses $C_1, \ldots, C_m$, where each clause is the disjunction of two literals. Examples of possible clauses are $x_i \vee x_j$, $x_k \vee \neg x_\ell$ and so on. The goal is to find a truth assignment to the $n$ variables so that all clauses are satisfied (if one exists).

The algorithm we are going to analyze is ingenious and simple:

1. Select uniformly at random a truth assignment $\pi$.
2. Repeat the following for at most $O(n^2)$ steps (or until we find a satisfying assignment):

   Select an unsatisfied clause $C_\ell$ with variables $x_i$, $x_j$.

   Select one of the variables $x_i$ and $x_j$ with equal probability.

   Update $\pi$ by flipping the truth assignment of the selected variable, i.e., set it to true if it was false and vice versa.

In the following, we assume for simplicity that there is a satisfying truth assignment $\pi^*$ (where $\pi_i^*$ denotes the Boolean value of variable $x_i$). (If there is no such truth assignment we cannot find one so there is nothing to prove.)

For a truth-assignment $\pi$, define $\text{dist}(\pi^*, \pi)$ as the number of variables/coordinates where $\pi_i \neq \pi_i^*$ (this is called the *Hamming distance*). Let $\pi = \pi^1, \pi^2, \ldots$ be the truth-assignments generated by the algorithm and let $d_i = \text{dist}(\pi^*, \pi^i)$. Note that as the algorithm only changes one variable in the truth assignment in each iteration, we have $d_{i+1} = d_i + \Delta_i$ where $\Delta_i \in \{-1, 1\}$. Prove that for any $i$

$$\Pr[\Delta_i = -1] \geq 1/2\,.$$

Once this is established, we can view the algorithm as a random walk on $\{0, 1, 2, \ldots, n\}$ that always goes to the left (to a smaller number) with probability at least $1/2$. It is known that such a random walk will reach 0 with high probability after $O(n^2)$ many steps. (Show this if you feel that you are up for a challenge.)