# Advanced Algorithms - Problem set 2

Whenever you give an algorithm, also argue for its running time and correctness. Always feel free to ask on MS Teams if you're stuck, need a clarification, or suspect a typo.

1. Some exercises on probability:
   (a) Say we want to model the random experiment of throwing $n$ unbiased coins. How many elements are contained in the sample space, and how many possible events are there?
   (b) Give probability spaces with events $A, B$ such that:
      i. $\Pr(A|B) > \Pr(A)$
      ii. $\Pr(A|B) = \Pr(A)$
      iii. $\Pr(A|B) < \Pr(A)$
   (c) Let $p_n$ be the probability that a string $x \in \{0,1\}^n$, chosen uniformly at random, does not contain two consecutive 1-entries. Show that $p_n \to 0$ as $n \to \infty$. (Hint: One particular way of showing this would split each string $x \in \{0,1\}^n$ into small pieces, determine the probability that each piece satisfies the desired property, and then use this to derive an upper bound of $O(c^n)$ for $c < 2$ on the number of $x \in \{0,1\}^n$ satisfying the property.)

2. Consider a random experiment that succeeds with probability at least $p$. The experiment can be repeated.
   (a) Give a lower bound on the probability of succeeding at least once in $t$ independent trials. (Hint: Consider the event that you fail in $t$ trials. You want to avoid this event.)
   (b) Depending on $p$, determine a number $c \in \mathbb{N}$ such that, with probability at least 0.99, the experiment succeeds at least once in $c$ trials. (Hint: This is just calculation.)
   (c) In the lecture, we analyzed Karger's algorithm and found that, in a graph with $n$ vertices, it succeeds in finding any fixed minimum cut with probability at least

$$\frac{2}{n(n-1)}.$$

   How many repetitions of Karger's algorithm are needed to make sure that, with probability at least 0.99, it succeeds at least once? Use that $1 - x \leq e^{-x}$ to show that the number of repetitions is $O(n^2)$.

3. You just finished your business trip to Zamazon and would like to collect business cards from its employees. Zamazon is a young start-up and thus extremely serious about portraying itself as fun, so it has meticulously designed an engaging and fun procedure for business card collection: The $n$ employees align in a queue in front of you. The first employee in the queue (counting from the back) passes his business card to the second employee. The second employee either passes the card he received to the third employee (with probability $\frac{1}{2}$) or discards the card and passes his own card to the third employee (with probability $\frac{1}{2}$). In general, the $k$-th employee either passes on the card he has received (with probability $1 - \frac{1}{k}$) or discards the card he received and passes on his own card (with probability $\frac{1}{k}$). You are the recipient of the card passed on by the $n$-th employee. All of this is synchronized to upbeat dance music.
   (a) After one run of this procedure, what is the probability that you receive the business card of employee $k$ for $k \in \{1, \dots, n\}$? Show how you derived the solution.
   (b) How many runs does this procedure run in expectation before you have received the business cards of all employees?

4. Let $G = (V, E)$ be a graph with $n$ vertices. For $k \in \mathbb{N}$, a *walk* of length $k$ in $G$ is a sequence of vertices $W = (v_0, \dots, v_k)$ such that $\{v_i, v_{i+1}\} \in E$ for all $0 \le i < k$. Note that the vertices are not required to be distinct. We say that a walk $W = (v_0, \dots, v_k)$ is a *path* if the vertices $v_0, \dots, v_k$ are all distinct.

   (a) Let $A$ be the adjacency matrix of $G$. For $k \in \mathbb{N}$, show that the entry $(u, v)$ of $A^k$ counts the walks of length $k$ from $u$ to $v$ in $G$. Using this, give an algorithm for counting walks of length in $k$ in time $O(n^\omega)$ when $k = O(1)$.

   (b) Let $c : V(G) \to \{1, \dots, k\}$ be a function. We view $c$ as a function that assigns a color to each vertex of $G$. A walk $W = (v_0, \dots, v_k)$ is *colorful* if $c(v_i) \neq c(v_j)$ for all $0 \le i < j \le k$. That is, every color appears exactly once in $W$. Give an algorithm for counting colorful walks of length in $k$ in time $O(n^\omega)$ when $k = O(1)$. (Hint: Since we assume that $k = O(1)$, you can enumerate all possible orderings of the $k$ colors along a path in $k! = O(1)$ time. For any ordering of colors, count paths that respect this ordering. To do this, rather than taking $A^k$, define appropriate matrices $B_{i,j}$ for $1 \le i, j \le k^2$ and take their products.)

   (c) Let $W$ be any fixed walk of length $k$ in $G$. We draw a coloring

   $$c : V(G) \to \{1, \dots, k\}$$

   uniformly at random.

      i. If $P$ is a path, then what is the probability that $W$ is colorful? (Hint: The walk $W = (v_0, \dots, v_k)$ has $k$ vertices. Count the good colorings on these vertices and divide by the number of possible colorings.)

      ii. Using the fact that

      $$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

      for any $x \in \mathbb{R}$, show that the above probability is at least $\frac{1}{e^k}$.

      iii. If $P$ is not a path, then what is the probability that $W$ is colorful? (Hint: This is a trick question.)

   (d) Using the above parts, give a randomized algorithm for testing whether $G$ contains a path of length $k$. Your algorithm should run in time $O(n^\omega)$ for $k = O(1)$. It should always give the right answer when $G$ has no path, and it should give the right answer with probability at least 0.99 when $G$ does contain a path.