CS 761 Randomized Algorithms **Last Updated:** May 21 (11am), 2018
Homework 1 **Due Date: June 1 (11pm), 2018**

You are allowed to discuss with others but not allowed to use any references except the course notes and the books "Probability and Computing" and "Randomized Algorithms". Please list your collaborators for each question. This will not affect your marks. In any case, you must write your own solutions.

There are totally 80 marks, and the full mark is 70.

---

1. **Minimum k-cut**

   (10 marks) Generalizing on the notion of a cut-set, we define a k-way cut-set in an undirected graph as a set of edges whose removal breaks the graph into k or more connected components. Show that the randomized contraction algorithm can be modified to find a minimum k-way cut-set in $n^{O(k)}$ time.

2. **Online Hiring**

   (10 marks) You need to hire a new staff. There are n applicants for this job. Assume that you will know how good they are (as a score) when you interview them, and the score for each applicant is different. So there is a unique candidate with the highest score, but you don't know that the applicant is the best when you interview him/her until you have interviewed all the applicants. The problem is that after you interview one applicant, you need to make an online decision to either give him/her an offer or forever lose the chance to hire that applicant. Suppose the applicants come in a random order (i.e. a uniformly random permutation), and you would like to come up with a strategy to hire the best applicant.

   Consider the following strategy. First, interview m applicants but reject them all. Then, after the m-th applicant, hire the first applicant you interview who is better than all of the previous applicants that you have interviewed.

   Let E be the event that you hire the best applicant. Let E_i be the event that the i-th applicant is the best and you hire him/her. Compute $\Pr(E_i)$ and show that

   $$\Pr(E) = \frac{m}{n} \sum_{j=m+1}^{n} \frac{1}{j-1}.$$

   For an appropriate choice of m, show that $\Pr(E)$ can get arbitrarily close to $1/e$ when n tends to infinity.

3. **Random Permutations**

   (10 marks) A permutation $\pi : [n] \to [n]$ can be represented as a set of cycles as follows. Let there be one vertex for each number i for $1 \leq i \leq n$. If the permutation maps the number i to the number $\pi(i)$, then a directed arc is drawn from vertex i to vertex $\pi(i)$. This leads to a graph that is a set of disjoint cycles. Notice that some of the cycles could be self-loops. What is the expected number of cycles in a random permutation of n numbers?

4. **Quicksort**

   (10 marks) The expected runtime of the randomized quicksort algorithm is at most $2n \ln n$ steps where n is the number of elements to be sorted. Prove that the probability that the actual runtime is more than say $100n \ln n$ is at most inverse polynomial in n.

5. **Chernoff Bound**

   (15 marks) Recall that X is a standard normal random variable if its probability density function is $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$.

   (a) Compute the moment generating function $M_{X^2}(t) := E[e^{tX^2}]$. You may use the fact that $\int_{-\infty}^{\infty} f(x) = 1$ without providing a proof.

   (b) Use (a), or otherwise, to compute $E[X^4]$.

   (c) Let $X_1, \ldots, X_k$ be independent standard normal random variables, and $Y = \frac{1}{k}\sum_{i=1}^{k} X_i^2$. Derive a Chernoff bound to prove that $\Pr(Y > 1 + \epsilon) \leq e^{-k\epsilon^2/8}$ for $0 \leq \epsilon \leq 1$. You may use the Taylor expansion $\ln(1 - x) = -\sum_{i=1}^{\infty} x^i/i$ for $-1 \leq x \leq 1$.

6. **Graph Sparsification**

   (10 marks) Given a complete graph G with n vertices where every edge is of the same weight, explain that Karger's uniform sampling algorithm will succeed with high probability in giving a $(1 \pm \epsilon)$-cut approximator H of G with H having only $O((n \log n)/\epsilon^2)$ edges.

   Also, provide a heuristic argument that when given the unweighted complete graph G as input, the uniform sampling algorithm requires $\Omega(n \log n)$ edges to produce any reasonable cut approximator H of G.

7. **Minimum Multicut**

   (15 marks) We are given a graph $G = (V, E)$ and k pairs of source-sink vertices, $s_i, t_i \in V$ for $i = 1, \ldots, k$. We wish to find a subset of edges $F \subseteq E$ that minimizes $|F|$ such that for each $i = 1, \ldots, k$, there is no s_i-t_i path in $(V, E - F)$.

   Consider the following vector program:

$$\begin{array}{lrll} \text{minimize} & \sum_{(i,j)\in E}(1 - v_i \cdot v_j) & & \\ \text{subject to} & v_{s_i} \cdot v_{t_i} & = 0, & i = 1, \ldots, k, \\ & v_j \cdot v_j & = 1, & \forall j \in V, \\ & v_j & \in \mathbb{R}^n, & \forall j \in V. \end{array}$$

   Consider the demand graph $H = (V, E')$, where $E' = \{(s_i, t_i) : i = 1, \ldots, k\}$. Let Δ be the maximum degree of a vertex in the demand graph. Suppose that the optimal value of the vector programming relaxation is $\epsilon|E|$. Consider the following algorithm. We draw $t = \lceil \log_2(\Delta/\epsilon) \rceil$ random unit vectors $r_1, \ldots, r_t$. The t random vectors define 2^t different regions into which the vectors v_i can fall: one region for each distinct possibility of whether $r_j \cdot v_i \geq 0$ or $r_j \cdot v_i < 0$ for all $j = 1, \ldots, t$. Remove all edges (i, j) from the graph such that v_i and v_j are in different regions. If for any s_i-t_i pair, there still exists an s_i-t_i path, remove all edges incident on s_i. We now analyze this algorithm.

   (a) Prove that the vector program is a relaxation of the minimum multicut problem.

(b) For any $(i, j) \in E$, prove that the probability that i and j are in different regions is at most $t \cdot \sqrt{1 - v_i \cdot v_j}$. You may use the fact that $1 - \cos(\theta) - 2\theta^2/\pi^2 \geq 0$ for $-\pi \leq \theta \leq \pi$ without providing a proof.

(c) Prove that for any $i = 1, \ldots, k$, the probability that we end up removing all the edges incident on s_i is at most $\Delta 2^{-t}$.

(d) Show that the expected number of edges removed is at most $O(\sqrt{\epsilon} \log(\Delta/\epsilon)|E|)$.

For the final item, it may be useful to use Jensen's Inequality, which states that for any convex function f and any positive p_i,

$$f\left(\frac{\sum_i p_i x_i}{\sum_i p_i}\right) \leq \frac{1}{\sum_i p_i} \sum_i p_i f(x_i).$$