CSE 521: Design and Analysis of Algorithms **Fall 2020**

Problem Set 1

Deadline: Oct 16 (at 11:59 PM) in Canvas

Instructions

- You should think about each problem by yourself for at least an hour before choosing to collaborate with others.
- You are allowed to collaborate with fellow students taking the class in solving the problems (in groups of at most 2 people for each problem). But you **must** write your solution on your own.
- You are not allowed to search for answers or hints on the web. You are encouraged to contact the instructor or the TAs for a possible hint.
- You cannot collaborate on Extra credit problems
- Solutions typeset in LATEX are preferred.
- Feel free to use the Discussion Board or email the instructor or the TA if you have any questions or would like any clarifications about the problems.
- Please upload your solutions to Canvas. The solution to each problem **must** be uploaded separately.

In solving these assignments and any future assignment, feel free to use these approximations:

$$1 - x \approx e^{-x}, \qquad n! \approx (n/e)^n, \qquad \left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$$

1) In this problem you are supposed to implement Karger's min-cut algorithm.

   I have uploaded three input files to the course website. Each file contains the list of edge of a graph; note that the graphs may also have parallel edges. The label of each node is an integer. For example, given the following input you should output 1. This graph has 4 edges and nodes have labels $1, 3, 4, 6$. The

   $$\begin{matrix} 1 & 3 \\ 3 & 4 \\ 4 & 6 \\ 6 & 3 \end{matrix}$$

   minimum cut is the degree cut of vertex 1.

   There are four inputs uploaded to the course website. For each input file you should output the size of the minimum cut. Please upload your code to Canvas. You should also write the output of your program for each input in the designated "text box" of Problem 1 in Canvas.

2) In the min s-t cut problem, we are given a graph $G = (V, E)$ and a pair of vertices s, t we want to find

   $$\min_{(S,\overline{S}): s \in S, t \in \overline{S}} |E(S, \overline{S})|,$$

   i.e., the smallest cut separating s from t. Consider executing a variant of the Karger's algorithm, in which each time we choose a uniformly random edge e, conditioned on the two (super) nodes which are endpoints of e do not contain both s, t and we contract e; in other words, we never merge s, t in the course of the algorithm. We repeat this until only two (super) nodes remain where one contains s and the other contains t. So, we just output the corresponding s-t cut.

In this exercise we want to show this algorithm fails, i.e., the probability of success is so small that we cannot turn it into a high probability success even by running polynomially many copies of this algorithm. Construct a graph G with n vertices such that the above algorithm outputs the min s-t cut with exponentially small probability, namely c^{-n} for some constant $c > 0$. For convenience, graph G may also have parallel edges.

3) Consider a permutation $p_1, \dots, p_n$ of numbers $[n] := \{1, \dots, n\}$. An increasing subsequence is a sequence $a_1 < a_2 < \cdots < a_k$ (for some $k \geq 1$) such that $p_{a_1} \leq \cdots \leq p_{a_k}$. For example, the longest increasing subsequence in $1, 3, 2, 4$ has length 3. Prove that the expected length of the longest increasing subsequence in a uniformly random permutation of numbers $[n]$ is at least $\sqrt{n}/c$ for some constant $c > 0$.

4) In lecture 4 we (will) discuss the pairwise independent hash functions. We say that for a prime p we can generate a pairwise independent hash function by choosing a, b independently from the interval $\{0, \dots, p-1\}$ and using $ax + b$ as a random number. Suppose we generate t pseudo random numbers this way, $r_1, \dots, r_t$ where $r_i = ai + b$. We want to say this set is far from being mutually independent. Consider the set $S = \{p/2, \dots, p-1\}$ which has half of all elements. Prove that with probability at least $\Omega(1/t)$ none of the pseudo-random-numbers are in S. Note that if we had mutual independence this probability would have been $1/2^t$.

5) Given a graph $G = (V, E)$ with $n = |V|$ vertices. Let k be the size of the minimum cut of G. In this problem we show that if k is large enough we can down sample G and preserve the size of its min-cut.

   Let H be a subgraph of G defined as follows: For every edge e of G include e in H with probability $p = \frac{12 \ln n}{k\epsilon^2}$. If we include e in H we weight it by $1/p$.

   We show that the (weighted) min-cut of H is at least $k(1-\epsilon)$ with probability at least $1 - 1/n$. Note that H has only $p|E|$ many edges (in expectation). So, H has significantly less edges than G if $k \gg \ln(n)/\epsilon^2$.

   a) For a cut $(S, \overline{S})$, let
   $$w_H(S, \overline{S}) = \frac{|H(S, \overline{S})|}{p}$$
   be the sum of the weights of all edges of H across this cut. Show that for any set $S \subset V$,
   $$\mathbb{E}\left[w_H(S, \overline{S})\right] = |E(S, \overline{S})|$$

   b) Use Chernoff bound to show that for any set $S \subset V$,
   $$\mathbb{P}\left[w_H(S, \overline{S}) < (1-\epsilon)|E(S, \overline{S})|\right] \leq e^{-6 \ln n |E(S, \overline{S})|/k} = n^{-6|E(S, \overline{S})|/k}$$

   c) Recall that in class we proved that each graph has at most $\binom{n}{2}$ many min-cuts. We say a cut of G is an α-min-cut if its size is at most α times the size of the min-cut, i.e., at most αk many edges. It follows by an extension of the Karger's algorithm that any graph G has at most $n^{2\alpha}$ α-min-cuts. Use union bound (and the latter fact) to show that for any integer $\alpha \geq 1$, with probability at least $1 - 1/n^2$, for all sets S where $\alpha k \leq |E(S, \overline{S})| \leq 2\alpha k$,
   $$w_H(S, \overline{S}) \geq (1-\epsilon)|E(S, \overline{S})|.$$

   d) Use another application of union bound to show that with probability at least $1 - 1/n$, for all sets $S \subset V$
   $$w_H(S, \overline{S}) \geq (1-\epsilon)|E(S, \overline{S})|.$$