CSE 521: Design and Analysis of Algorithms **Fall 2023**

Problem Set 1

Deadline: Oct 09 (at 11:59 PM) in gradescope

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. 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 TAs if you have any questions or would like any clarifications about the problems.
- Please upload your solutions to Gradescope.

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

$$1 - x \approx e^{-x}, \qquad \sqrt{1-x} \approx 1 - x/2, \qquad n! \approx (n/e)^n, \qquad \left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$$

1) Given a graph $G = (V, E)$ with $n = |V|$ vertices, a cut is an α-approximate min cut, if the number of its edges is at most α times the minimum cut of G. Use an extension of the contraction algorithm to show that any graph G has at most $n^{2\alpha}$ many α-approximate min cuts. You would also receive full credit if you show that the number of α-approximate min cuts is at most $n^{O(\alpha)}$.

2) Consider adapting the min-cut algorithm to the problem of finding an s-t min-cut in an undirected graph. In this problem, we are given an undirected graph $G = (V, E)$ together with two distinguished vertices s and t. An s-t cut is a set of edges whose removal from G disconnects s from t; we seek an s-t cut of minimum cardinality. As the algorithm proceeds, the vertex s may get amalgamated into a new super-node as a result of an edge being contracted; we call this vertex the s-vertex (initially the s-vertex is s itself). Similarly, we have a t-vertex. As we run the contraction algorithm, we ensure that we never contract an edge between the s-vertex and the t-vertex, i.e., among all edges which are not between the s-vertex and the t-vertex we choose one uniformly at random. Show that there are graphs in which the probability that this algorithm finds an s-t min-cut is very small! How small this probability can be as a function of $n = |V|$, the number of vertices of G?