CS 671: Graph Streaming Algorithms and Lower Bounds **Rutgers: Fall 2020**

Problem set 1

Due: 11:59PM, September 15, 2020

Problem 1. Design a single-pass semi-streaming algorithm for *one* of the following two problems (you can decide whichever one you would like to solve):

(i) Finding a minimum spanning tree (MST) of the input *weighted* undirected graph $G = (V, E)$ where the weight of each edge is revealed at the same time with the edge in the stream. You may assume that G is connected and all edge-weights are distinct, thus MST of G is unique.

(ii) Given a *constant* $k > 0$, deciding whether the input undirected graph $G = (V, E)$ is k-*vertex*-connected or not. This is similar to the k-edge-connectivity problem in Lecture 1 with the difference that we now want to check whether deleting at most k *vertices* can make the remaining graph disconnected or not.

Problem 2. In the *set disjointness* communication problem, Alice and Bob are given subsets $A, B \subseteq [N]$, respectively. The goal is to determine whether or not $A \cap B = \emptyset$.

(i) Prove that the deterministic communication complexity of this problem is $\mathrm{D}(\text{set-disjointness}) = \Omega(N)$.

Remark. Note that unlike in Lecture 1, here, the goal is to prove a lower bound for two-way communication complexity. You may want to check the following notes from CS 514 for some pointers on proving communication complexity lower bounds:

`https://www.cs.rutgers.edu/~sa1497/courses/cs514-s20/lec8.pdf`

Remark. One can in fact prove that even randomized communication complexity of this problem is $\mathrm{R}(\text{set-disjointness}) = \Omega(N)$; we will hopefully get to this result later in the semester.

(ii) Use this in a reduction to prove that any *constant pass* deterministic streaming algorithm for undirected connectivity requires $\Omega(n)$ bits of space.