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

Problem set 12

Due: 11:59PM, December 8, 2020

Problem 1. In Lecture 13, we saw a single-pass semi-streaming $(3/2+\varepsilon)$-approximation algorithm for the maximum matching problem in *random-arrival streams*. In this question, we try to extend this result to another streaming arrival model. Consider the following setting, known as *random vertex arrival*:

- The adversary picks an arbitrary bipartite graph $G = (L, R, E)$ with $|L| = |R| = n$.
- Let $v_1, \ldots v_n$ be a *random permutation* of the vertices in L.
- The edges arrive in the following order: first all edges incident to v_1 in *an adversarial order*, then all edges incident to v_2, and so on.

Show how the algorithm from Lecture 13 can be modified to achieve a $(3/2+\varepsilon)$-approximation to maximum bipartite matching in this random vertex arrival model.

Hint: Firstly, you will need to adjust many of the parameters. More importantly, there are two bigger changes you will need to make: (i) You will want to change the termination condition of an epoch, and (ii) you will sometimes want to change H by many edges in a single epoch, but here you have to be careful: just because an epoch contains many *underfull* edges, does not directly imply that you can make many changes to H, since it is possible that adding a single underfull edge makes all the other edges not underfull. To compensate, you will want to use the fact that there is slack between our definition of underfull (edge-degree less than $\beta(1-\lambda)$) and how we defined deletion moves (edge-degree less than $\beta - 1$)[1].

[1]Many thanks to Aaron Bernstein for formulating this wonderful question and his hint for solving it.