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

Problem set 10

Due: 11:59PM, November 17, 2020

Problem 1. Given a graph $G = (V, E)$, a MIS of G is any independent set (vertices with no edges to each other) which is maximal (any other vertex is neighbor to at least one of the chosen vertices). We design an $O(\log\log n)$-pass semi-streaming algorithm for the maximal independent set (MIS) problem in this question.

Consider the following algorithm:

1. Let $v_1, \ldots, v_n$ be a fixed ordering of vertices of G.
2. Sample every *vertex* of G independently and with probability $p > 0$. Let S be this sample and $G[S]$ be the *induced* subgraph of G on these vertices.
3. Let $T \leftarrow \emptyset$ and iterate over S in this order and for each vertex $v_i \in S$: if no neighbor of v_i in S belongs to T, set $T \leftarrow T \cup \{v_i\}$.

Let H denote the graph obtained after removing T *and* all vertices incident on T from G. Observe that the output T of this algorithm is an independent set in G but not necessarily a maximal one.

(*i*) Prove that with high probability, maximum degree of H is $O(\frac{1}{p} \cdot \log n)$.

(*ii*) Prove that with high probability, number of edges in $G[S]$ is $O(pn \cdot p\Delta \cdot \log n)$, where Δ is the maximum degree of G.

Let us now use this subroutine to solve the MIS problem.

1. Let $\mathcal{M} \leftarrow \emptyset$, and $G_1 = G$. For $i = 1$ to ∞:
   (a) If G_i has $O(n \log n)$ edges, store G_i explicitly and **return** an MIS of G_i plus $\mathcal{M}$.
   (b) Run the algorithm above with parameter $p_i = n^{-1/2^i}$ on the graph G_i to obtain a set T_i.
   (c) Let $\mathcal{M} \leftarrow \mathcal{M} \cup T_i$ and let G_{i+1} be the graph obtained after removing T_i and all vertices incident on it from G_i.

(*iii*) Prove that the for-loop in this algorithm runs for at most $O(\log\log n)$ passes and the algorithm computes an MIS of the input graph.

(*iv*) Prove that with high probability, this algorithm only requires $O(n \cdot \text{polylog}\,(n))$ space.