6.S899 Distributed Graph Algorithms

Problem Set 2

Due date: 10/31/2014

1 Randomized Ruling Set

In lecture 4, we saw a deterministic algorithm for computing a $(k, k \log n)$-Ruling Set in $O(k \log n)$ rounds. Here, we analyze a simpler and faster but randomized algorithm.

Let each node v pick a random $6 \log n$-bit number $r_v \in \{0,1\}^{6 \log n}$. Then, make v join the set S if the random number r_v of v is strictly larger than the random number r_u of all neighbors u of v.

Clearly S is an independent set. Show that with probability at least $1 - 1/n$, for each node v, there is at least one node $s \in S$ within $O(\log n)$ hops of v. Also explain how to extend this idea to an algorithm for generating a $(k, O(k \log n))$-Ruling Set.

2 Minimum Spanning Tree (MST) of a Subgraph

In lecture 5 we showed that a MST of a graph G can be computed in time $\tilde{O}(\sqrt{n}+D)$. Now assume there is a connected subgraph H of G and we want to compute an MST T_H of H, but are allowed to use communication in G (not only communication in H). How fast can you compute an MST T_H? The argument should take only a few lines.

Hint: Let D_H be the diameter of H. The best time is one of the following:

$$\tilde{O}(\sqrt{n} + D_H) \quad , \quad \tilde{O}(\sqrt{n \cdot D_H} + D) \quad , \quad \tilde{O}(\sqrt{n} + D).$$

3 A Lower Bound for Minimum Cut Approximation

In lecture 6 we showed approximation lower bounds for MST and Single Source Shortest Paths (SSSP). Now we consider the problem of approximating a Minimum Cut (see definition below). Prove that $\alpha(n)$-approximating a minimum cut takes $\tilde{\Omega}(\sqrt{n} + D)$ rounds.

Definition: A set of edges $E' \subseteq E$ is a cut of G if G is not connected when we delete E'. The minimum cut problem is to find a cut of minimum weight. A cut E'' is an $\alpha(n)$-approximation to a Minimum Cut E', if

$$\omega(E') \leq \omega(E'') \leq \alpha(n) \cdot \omega(E').$$