

Lecture 12

Lecturer: Mohsen Ghaffari

Scribe:

Exercise 12

Pipelining

- (1a) The algorithm idea is as follows. First, we find a BFS rooted at v that spans the whole graph, e.g., by broadcasting the smallest id and every node choosing a node along the shortest path to v as its parent. Notice that while broadcasting, the first neighbor forwarding the id if v is along the shortest path to v . After the BFS is constructed, each leaf node u can start sending values $x_1(u), \dots, x_k(u)$ towards the root, one per round. Upon receiving values, every node only stores the minimum of the corresponding entry in $x_1(u), \dots, x_k(u)$ and forwards it towards the root in the following round. Once the leader has received all the minimum values, it can broadcast the values back to all nodes. Given that forwarding the values up and down the BFS tree does not suffer from congestion, the time complexity is bounded by the time needed for constructing the BFS tree and the height of the tree, yielding $O(D + k)$.
- (1b) Consider any two nodes v and u in distance d from each other, where u initially holds message m . The goal is to show by induction, that either m reaches v quickly (and is forwarded), or v forwards many other messages.

More formally, assume that for any $t = d + \ell$, after t rounds, node u has either sent message m or ℓ other messages. The base cases where either $t = 0$ or $d = 0$ follow from observing that there are no requirements in the case of $t = 0$ and no congestion in the case of $d = 0$.

Suppose then that the claim holds for $d - 1$ and $t - 1$ and let v' be the node adjacent to v along the shortest path from u to v . By the induction hypothesis, by the end of round $t - 1$, node v has either sent m or $\ell - 1$ other messages. Notice that distance from u to v' is $d - 1$ and therefore, node v' has either sent m or $(t - 1) - (d - 1) = t - d = \ell$ other messages.

In either case, node v will sent either m or some other message in round t proving the claim.

Minimum Spanning Tree

- (2a) Since every node selects a proposal edge and every proposal edge can be selected by at most two nodes, we get that at least $n/2$ distinct proposal edges are selected. Since the coin tosses are independent, we get that any proposal edge is marked with probability $1/4$. Let P be the set of proposal edges, where $|P| \geq n/2$ and let $X = \sum_{e \in P} X_e$, where X_e is an indicator random variable that gets value one if e is marked and zero otherwise. Given the above observations we can bound $\mathbb{E}[X] \geq |P| \cdot (1/4) \geq (n/2)/4 = n/8$.
- (2b) For every contraction, at least one node is “removed” from the graph as a result of the contraction. Thus, the expected number of remaining nodes is at most $n - \mathbb{E}[X] \leq (7/8)n$.
- (2c) By exercise (2b), we get that after $20 \log n$ iterations, the expected number of nodes remaining in the graph is bounded from above by

$$n \cdot \frac{7^{20 \log n}}{8} + 1 = n \cdot \frac{7^{\frac{20 \log_{7/8}(n)}}{\log_{7/8}(2)}}}{8} + 1 \leq n \cdot \frac{7^{-3 \log_{7/8}(n)}}{8} + 1 \leq n \cdot n^{-3} + 1 \leq \frac{1}{n^2} + 1$$

By Markov's inequality, the probability that more than one node remains after $20 \log n$ iterations is therefore bounded from above by $(n^{-2})/1 = n^{-2}$.

References

1. Distributed Broadcast Revisited: Towards Universal Optimality. Mohsen Ghaffari 42nd International Colloquium on Automata, Languages, and Programming, July 2015.