ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2011

Prof. R. Wattenhofer
Barbara Keller

Principles of Distributed Computing
Exercise 13

1 Lightest Edges

In this exercise, we use the all-to-all communication model in which each node can communicate with all other $n-1$ nodes in each synchronous communication round, i.e., the communication graph is the complete graph $\mathcal{K}_n$. Each node v has a unique identifier id_v and each edge e in the communication graph has a positive weight $w(e)$. You can assume all edge weights to be unique. The size of any message is restricted in that only a constant number of node identifiers, edge weights, and additionally a constant number of other numbers of the same magnitude can be sent in a single round.

In the lecture, we discussed algorithms to compute the minimum spanning tree (MST) in this model. Now we are interested in finding the n lightest edges overall, i.e., after the algorithm terminates, every node knows the weights of the n lightest among all $\binom{n}{2}$ edges and which nodes these edges connect. Initially, each node v knows the weights of all incident edges.

Consider the following simple algorithm:
Every node sends its i^{th} lightest edge to all other nodes in round i. After a sufficiently large number of rounds, the algorithm terminates and each node knows that the n smallest weights it has learnt belong to the n lightest edges overall.

a) Show an example (that means, an assignment of edge weights to the nodes) where the above algorithm is as slow as possible!
Hint: The problem with this simple algorithm is that nodes potentially send edge weights that have already been broadcast (by other nodes) before.

In order to overcome the problem mentioned above, we modify the algorithm in the following way: *In each round, broadcast the lightest incident edge weight that has not already been broadcast before.*

b) Prove an upper bound on the number of rounds required when the modified algorithm is used! Moreover, prove that your bound is asymptotically tight by providing a worst-case example (of the same asymptotic time complexity)!

Now, we are going to derive a *randomized* algorithm whose *expected* time complexity is only $O(1)$. Use the fact that a single node can determine the n^{th} smallest among all $\binom{n}{2}$ edge weights in $O(1)$ rounds in expectation.[1]

c) Given that node v knows the n^{th} smallest edge weight (after $O(1)$ rounds), how can all nodes (v and all other nodes) learn all the weights of the n lightest edges? Describe an algorithm that solves this problem! The total time complexity must not exceed $O(1)$ rounds!

[1] Note that this algorithm cannot be *parallelized*! This means that this subroutine cannot be used to compute all n lightest edges *in parallel* in $O(1)$ time.