1 Pipelining

We consider an arbitrary \( n \)-node network \( G = (V, E) \) with diameter \( D \). Moreover, we work in the CONGEST model of distributed computing where each node has an \( O(\log n) \)-bit unique identifier and per round, each node can send \( O(\log n) \) bits to each of its neighbors.

Exercises

(1a) Suppose that each node \( v \in V \) is given \( k \) different inputs \( x_1(v), x_2(v), \ldots, x_k(v) \), each being a \( \Theta(\log n) \)-bit number. The objective is to for all nodes to know the outputs \( y_i = \min_{v \in V} x_i(v) \), for each \( i \in \{1, 2, \ldots, k\} \). Devise a deterministic distributed algorithm for this problem with round complexity \( O(D + k) \).

(2a) Suppose there are \( k \) messages \( m_1, m_2, \ldots, m_k \), each initially placed at an arbitrary node of the network (many or even all of the messages may be placed on the same node). Consider the following basic algorithm: per round, each node \( v \) picks one of the messages \( m_i \) that it has (from the beginning or received in the past) and sends \( m_i \) it to all of its neighbors; node \( v \) will never send \( m_i \) again. Notice that a node will not send two of the messages at the same time. Prove that if we run this algorithm for \( O(D + k) \) rounds, all nodes will receive all the messages.

2 Minimum Spanning Tree

Consider an undirected connected graph \( G = (V, E) \) where \( n = |V| \). Suppose that each node \( v \in V \) has selected one of its incident edges \((v, u)\) as the proposal edge of \( v \), let us denote it \( e_v = (v, u) \). For instance, in the MST algorithm of Boruvka, this would be the minimum-weight edge incident on \( v \). Notice that the two endpoints of an edge might propose this one edge simultaneously. Consider the random process that each node flips a fair coin for itself and then, we mark the proposed edge \( e_v = (v, u) \) of node \( v \) only if \( v \) draws tail and \( u \) draws head.

Exercises

(2a) Prove that, in expectation, we mark at least \( n/8 \) edges.

(2b) Prove that, if we contract all the marked edges, the resulting graph has at most \( 7n/8 \) nodes, in expectation.

(2c) Consider repeating the above process for \( 20 \log n \) iterations: In each iteration, we contract all the marked edges, and maintain only the “outgoing edges”, i.e., those edges that have exactly one endpoint in this contraction. Then, select one min-weight outgoing edge per new node, and repeat the marking process as above using one coin toss per each new node. Use (2b) to prove that, after \( 20 \log n \) iterations, with high probability, we have contracted everything to a single node.