Principles of Distributed Computing
Sample Solution to Exercise 7

1 Lightest Edges

a) Clearly, the execution of this algorithm cannot take more than \( n \) rounds. Let the \((n - 1)\) lightest edges form two stars of the same size and the \( n^{th} \) lightest edge connect the two centers of the stars. We are not interested in the distribution of the other weights. In this scenario it takes 1 round until the \( n - 1 \) lightest edges are announced. In the next \( \lceil n/2 \rceil - 2 \) rounds, the centers of the stars will announce these edges once more. Only in round \( \lceil n/2 \rceil \) the two center nodes announce the \( n^{th} \) lightest edge. Since it is necessary to know this edge, the algorithm cannot terminate earlier and the time complexity of this algorithm is \( \Omega(n) \).

b) We first prove that the time complexity is upper bounded by \( \lceil 2\sqrt{n} \rceil \in O(\sqrt{n}) \). After \( \lceil 2\sqrt{n} \rceil \) rounds, all nodes with at most \( 2\sqrt{n} \) edges among the \( n \) lightest edges have broadcast all relevant edges known to them. That means, after \( \lceil 2\sqrt{n} \rceil \) rounds, there can only be missing edges between nodes that initially had at least \( 2\sqrt{n} + 1 \) lightest edges leading to nodes that are also incident to at least \( 2\sqrt{n} \) lightest edges. Assume there is such a node. Since each edge connects two nodes, initially we must have had at least \((2\sqrt{n}+1)\sqrt{n}/2 > n\) lightest edges, a contradiction.

We now construct a worst-case example. Each edge connecting two nodes from a specific set of \( \lceil 2\sqrt{n} \rceil \) nodes is assigned one of the \( n \) smallest weights. Since there are \( \binom{\lceil 2\sqrt{n} \rceil}{2} \leq n \) edges between these nodes, we know that all edges between these nodes must be broadcast. Apparently, the \( \lceil 2\sqrt{n} \rceil \) nodes will announce at most the same number of edges in each round. Thus, in total at least \( \lceil 2\sqrt{n} \rceil / 2 \in \Omega(\sqrt{n}) \) rounds are required.

c) Node \( v \) can send the \( n^{th} \) smallest edge weight to all nodes. Every node \( v_i \) can now determine how many among its edges \((v_i, v_j)\), where \( i < j \), belong to the \( n \) lightest edges and send this value \( N_i \) to all nodes. Now, the nodes know to which node they have to send their edge weights such that they can be distributed in the next round without contention: Node \( v_i \) sends its smallest weight to the node \( v_k \), where \( k = 1 + \sum_{j=1}^{i-1} N_j \), the next one to \( v_{k+1} \), etc. Thus, every node receives exactly one edge weight to forward to all nodes. This procedure takes four rounds, i.e., the time complexity is \( O(1) \).