University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, A. Mályusz, Z. Parsaeian

Theory of Distributed Systems
Sample Solution Exercise Sheet 3

In all the exercises of this problem set, we consider the synchronous message passing model on a graph, where nodes operate in synchronous rounds and all nodes start a computation together at time 0. We assume that initially, nodes do not know the IDs of their neighbors. Note that we will always denote by n the number of nodes in the graph and m the number of edges in the graph.

Exercise 1: Leader Election in General Graphs

Consider the following leader election algorithm. For simplicity, we assume that every node knows the graph diameter D. Every node u stores the largest ID it has seen in variable x_u. Each node $u \in V$ carries out the following algorithm.

Node u initially sets x_u :=ID(u) and sends x_u to its neighbors
for $D - 1$ rounds **do**
  if $x_v > x_u$ for the largest value x_v that u received **then**
    u sets $x_u := x_v$ and sends x_u to all neighbors from which it has not received a value equal to x_v
  end if
end for

After D rounds, the value x_u of each node u equals the largest ID in the network.

What is the message complexity of this algorithm (in terms of n)? Give an example that shows that your given bound is asymptotically tight (in the worst case) i.e. if $B(n)$ is your message complexity, give a family of graphs where your algorithm indeed has a message complexity of $\Omega(B(n))$.

Sample Solution

In every round of the algorithm, at most $2m$ messages are sent (at most one in each direction via each edge). The message complexity is therefore $O(D \cdot m)$, which is $O(n^3)$ in the worst case.
To show that this bound is tight in the worst case, we construct for a given n a graph, where $\Omega(n^3)$ messages are sent, in the following way: nodes with IDs $n, \ldots, \lceil n/2 \rceil$ form a path, nodes with IDs $\lceil n/2 \rceil, \ldots, 1$ form a clique. Let v be the node with ID $\lceil n/2 \rceil$. For $\lceil n/2 \rceil - 1$ rounds, v updates x_v and forwards it to its $\lceil n/2 \rceil - 1$ neighbors in the clique, which in the following round update their values and forward them to $\lceil n/2 \rceil - 2$ nodes (every node in the clique except themselves and v). So for $\lceil n/2 \rceil - 1$ rounds, there are at least $(\lceil n/2 \rceil - 1)(\lceil n/2 \rceil - 2)$ messages sent per round.

Exercise 2: Leader Election via Radius Growth

We generalize the radius growth algorithm for leader election from the lecture to arbitrary graphs. Assume that every node knows the number of nodes in the graph. The algorithm consists of phases $i = 0, 1, 2, \ldots$. Let C_i be the set of leader candidates at the beginning of phase i. Set $C_0 = V$ (initially, each node is a leader candidate). Phase i of the algorithm consists of 2^i rounds. The algorithm terminates when $2^i \geq n$. In phase i, each node $u \in V$ carries out the following algorithm.

If $u \in C_i$, u initializes $x_u :=$ID(u) and sends x_u to its neighbors (otherwise, u initializes $x_u := -1$).
for $2^i - 1$ rounds **do**
  if $x_v > x_u$ for the largest value x_v that u received **then**
    u sets $x_u := x_v$ and sends x_u to all neighbors from which it has not received a value equal to x_v
  end if
end for
if $u \in C_i \ \wedge \ x_u ==ID(u)$ **then**
  u joins C_{i+1} (i.e., u stays a candidate)
end if

a) Show that the number of messages sent in phase i is $O(\min\{2^i, |C_i|\} \cdot m)$.

b) Show that $|C_i| \le \frac{4n}{2^i}$ for each phase i.

c) Show with a) and b) that the message complexity of the algorithm is at most $O(m\sqrt{n}\log n)$.

d) For $m = \Omega(n^2)$, the upper bound from c) becomes $O(n^{5/2}\log n)$. Give an example network on which the algorithm requires $\Omega(n^{5/2})$ messages.

Sample Solution

a) On one hand, phase i consists of 2^i rounds and in one round at most $2m$ messages are sent, so there are at most $O(2^i \cdot m)$ messages.
On the other hand, in each phase only the values of nodes in C_i are broadcasted, each edge transports such a value at most once in each direction and we have $|C_i|$ such values, then that yields an upper bound of $O(|C_i| \cdot m)$ messages.

b) In phase i, two candidates $u, v \in C_i$ have distance at least 2^{i-1}, as otherwise the one with the smaller ID would have lost its candidate status at the end of phase $i-1$.
Consider the balls with radius $2^{i-2} - 1$ around u and v i.e. $\mathrm{B}_{2^{i-2}-1}(u)$ and $\mathrm{B}_{2^{i-2}-1}(v)$. They are disjoint (no node is contained in two different balls). Each of these balls contains at least 2^{i-2} nodes. It follows that $n \ge |C_i| \cdot 2^{i-2}$.

Recall definition: for a vertex u of graph G and an integer r, the ball of radius r centered at u denoted $B_r(u)$ is the subgraph induced by the set of all vertices of G whose distance from u does not exceed r.

c) It is enough to show that $\min\{2^i, |C_i|\} \in O(\sqrt{n})$ for every phase i to prove our message complexity. We notice that in each phase i, if $2^i \le |C_i|$, then $\min\{2^i, |C_i|\} = 2^i \le |C_i| \le \frac{4n}{2^i}$ (from part b), thus $(2^i)^2 \le 4n$, then $2^i \le 2\sqrt{n}$. And if $|C_i| \le 2^i$, then $\min\{2^i, |C_i|\} = |C_i| \le 2^i \le \frac{4n}{|C_i|}$ (from part b), thus $|C_i|^2 \le 4n$, then $|C_i| \le 2\sqrt{n}$. Hence in both cases, $\min\{2^i, |C_i|\} \le 2\sqrt{n}$, which shows what we want.

d) For a given n, we construct a graph with n nodes in the following way. For $i = 1, \ldots, \sqrt{n}$, there is a line such that the number of nodes on this line is i: (v_1^1), (v_1^2, v_2^2), (v_1^3, v_2^3, v_3^3), $\ldots$, $(v_1^{\sqrt{n}}, v_2^{\sqrt{n}}, \ldots, v_{\sqrt{n}}^{\sqrt{n}})$. We have now used up $\sum_{i=1}^{\sqrt{n}} i = \frac{n}{2} + \frac{\sqrt{n}}{2}$ nodes. So, we add $n - \sum_{i=1}^{\sqrt{n}} i = \frac{n}{2} - \frac{\sqrt{n}}{2} \in \Omega(n)$ additional nodes and together with the nodes v_i^i, they will form a clique. For each i, node $v_1^{\sqrt{n}-i}$ has the i^{th}-largest ID in the graph.

When running the algorithm on this graph, there are at least $\sqrt{n}$ rounds in which an ID reaches the clique which is larger than the ones it has seen before. Then all $\Omega(n)$ nodes v in the clique update their value x_v and send it to their $\Omega(n)$ neighbors in the following round. So there are at least $\sqrt{n}$ rounds with a message complexity of $\Omega(n^2)$.

Exercise 3: Leader Election in Complete Graphs

In a complete graph, one can trivially solve leader election in one round if every node sends its ID to all its neighbors. This requires $\Omega(n^2)$ messages. The following algorithm uses less messages at the cost of a slightly higher time complexity.

The algorithm consists of phases $i = 1, 2, \dots$. Let C_i be the set of leader candidates at the beginning of phase i. Set $C_1 = V$ (initially, each node is a leader candidate). In phase i, each node $u \in V$ carries out the following algorithm.

---

if $u \in C_i$ **then**
  u sends a probe message (may I be a leader?) containing its ID to $\min\{2^i, n-1\}$ arbitrary neighbors.
end if
Let v be the node with the largest ID from which u received a probe message
if $\text{ID}(v) > \text{ID}(u)$ **then**
  u sends back an acknowledgement to v
end if
if u received 2^i acknowledgements **then**
  u joins C_{i+1} (i.e., u remains a candidate)
end if

---

a) Argue that the algorithm solves leader election and analyze its time complexity.

b) Show that $|C_i| \le \frac{n}{2^{i-1}}$ for each phase $i \ge 1$.

c) Analyze the message complexity.

Sample Solution

a) The node with maximum ID remains candidate in every phase. In phase $\lceil \log n \rceil$ all candidates send probes to all other nodes and thus the node with maximum ID is the only one surviving. As a phase consists of two rounds, the runtime is $O(\log n)$.

b) For $u \in C_i$, let $A_u \subseteq V$ be the set of nodes from which u received an acknowledgement. We have $|A_u| = 2^{i-1}$ (from phase $i-1$) and for $u, v \in C_i$ we have $A_u \cap A_v = \emptyset$. Therefore, $n \ge |C_i| \cdot 2^{i-1}$.

c) In phase i except the last, there are at most $2 \cdot |C_i| \cdot 2^i$ messages sent and at most $2 \cdot |C_i| \cdot (n-1)$ messages for the last phase. With b) it follows that the message complexity of each phase is $O(n)$. As the algorithm has $\log n$ phases, its message complexity is $O(n \log n)$.