ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2016

Prof. R. Wattenhofer
Pankaj Khanchandani

Principles of Distributed Computing
Exercise 7: Sample Solution

1 Deterministic Maximal Independent Set

a) Consider the graph consisting of a connected chain of k nodes $v_1, \ldots, v_k$. We add $i-1$ additional edges leading to $i-1$ additional nodes at each node v_i for all $i \in \{1, \ldots, k-1\}$ and k additional nodes and edges to node v_k. The degree $\delta(v_1)$ of v_1 is 1 and for all other nodes $v_i \in \{2, \ldots, k\}$ we have that $\delta(v_i) = i + 1$. All additional nodes have degree 1.

In the first round, all nodes except v_k have a neighbor with a larger degree, thus only v_k joins the MIS. Afterwards, v_{k-1} can decide, then v_{k-2} and so on. Thus, after k time all nodes $v_1, \ldots, v_k$ and also the additional nodes have decided to join or not to join the MIS.

The number of nodes in this graph is

$$n = k + \sum_{i=1}^{k}(i-1) + 1 = 1 + \sum_{i=1}^{k} i = 1 + \frac{k(k+1)}{2} < \frac{(k+1)^2}{2}.$$

The time complexity is thus $k \geq \sqrt{2n} - 1 \in \Omega(\sqrt{n})$.

b) We first show that the above lower bound for trees is tight. Assume that there is a node v_0 of degree $\delta(v_0)$ which is still undecided at time $2\sqrt{n}$. Since v_0 is undecided, it must have a neighbor v_1 with higher degree, i.e., $\delta(v_0) < \delta(v_1)$, which was still undecided at time $2\sqrt{n} - 1$. As v_1 was undecided, it must in turn have a neighbor v_2 with a higher degree that was undecided at time $2\sqrt{n} - 2$. By induction it follows that there must be a chain of nodes $v_0, \ldots, v_{2\sqrt{n}}$, such that nodes v_i and v_{i+1} are neighbors and $\delta(v_i) < \delta(v_{i+1})$.

Note that for every neighbor x of a node v_i (assuming that x is not a node from the chain $v_0, \ldots, v_{2\sqrt{n}}$), that x cannot be a neighbor of another node v_j $(i \neq j)$, as otherwise there would be cycle and the graph would not be a tree. Analogously, a node v_i can only be connected to v_{i-1} and v_{i+1} and not to any other node v_j. Therefore, we can bound the total number of nodes in the tree using degree of the nodes of the chain: For every node v_i, there must be at least $\delta(v_i) - 2$ nodes that are only a neighbor of v_i. In addition, there are the $2\sqrt{n}$ nodes on the chain v_i themselves. Thus, the total number of nodes must be at least $2\sqrt{n} + \sum_{i=0}^{2\sqrt{n}} \delta(v_i) - 2$.

To establish a lower bound on the number of nodes required for this tree, we want to minimize this sum, which we can achieve by choosing the degrees as small as possible. Observe that the smallest degree $\delta(v_{2\sqrt{n}})$ must be 1, as it is connected to the chain. Since the degrees must be increasing, it follows that $\delta(v_{i-1}) = \delta(v_i) + 1$. Therefore

$$
\begin{aligned}
2\sqrt{n} + \sum_{i=0}^{2\sqrt{n}} \delta(v_i) - 2 &= \sum_{i=0}^{2\sqrt{n}} \delta(v_i) - 1 \\
&= \sum_{i=0}^{2\sqrt{n}} (i+1) - 1 \\
&= \sum_{i=0}^{2\sqrt{n}} i \\
&= \frac{2\sqrt{n}(2\sqrt{n}+1)}{2} \\
&= 2n + \sqrt{n} \\
&> 2n
\end{aligned}
$$

As we initially assumed that we have a graph with n nodes, and we now showed that for a runtime of $2\sqrt{n}$ we require more than $2n$ nodes, we reached a contradiction.

To construct a lower bound for general graphs we now consider a ring of k nodes $v_1, \ldots, v_k$ instead of a chain. We use $k-1$ additional nodes $u_1, \ldots, u_{k-1}$ to increase the degrees of the nodes v_i: There is an edge $\{v_i, u_j\}$ from all nodes v_i to all nodes u_j for which $j \in \{1, \ldots, k-i\}$. It is easy to see that the degree $\delta(v_i)$ of node v_i is $k+2-i$, and that $\delta(u_j) = k-j$.

In the first round, only v_1 joins the MIS. This means that all nodes $u_1, \ldots, u_{k-1}$ and also v_2 and v_k can no longer join the MIS. Thus, in the second round, all these nodes broadcast to all their neighbors that they will not join the MIS. In the third round, only v_3 decides to join the MIS because all other undecided nodes have an undecided neighbor with a larger degree. Subsequently, only v_4 decides (not to join the MIS) in round 4. Repeating this argument, we get that the last node v_{k-1} makes its decision not before round $k-1$. Since $n = k + (k-1)$, the time complexity is thus $k - 1 = \frac{n-1}{2} \in \Omega(n)$.

2 (Local) Reductions

a) We use Algorithm 36, which creates a coloring with the help of a MIS. As this algorithm would give us a node coloring and not an edge coloring, we preprocess the graph; i.e., we create the *line graph*. The line graph $L(G) = (E, F)$ has the edges from G as nodes, and has an edge between two nodes (edges of G) iff the edges share a node in G (we call these edges *adjacent*). Formally: $F = \{\{e, f\} \in \binom{E}{2} \mid e \cap f \neq \emptyset\}$. By construction of the line graph, it follows that a node coloring in the line graph $L(G)$ corresponds to an edge coloring in G. Observe that the number of nodes in $L(G)$ is in $O(n^2)$, and that the maximum degree in $L(G)$ is at most $2(\Delta - 1)$ (an edge may be adjacent to at most $\Delta - 1$ other edges at each of its nodes in G). Thus, the algorithm needs $O(\log(n^2)) = O(\log n)$ time w.h.p. and produces a coloring with at most $2(\Delta-1)+1 = 2\Delta - 1$ colors.

Let us now briefly discuss some details involved in the construction. The line graph can be simulated locally, where nodes of the line graph (i.e., edges of G) are simulated by one of their incident nodes. The nodes simulating adjacent edges are connected by them and therefore at most 2 hops away from each other. Thus, two rounds and (at most) two messages are required to simulate one round of communication and one message on the line graph, respectively. Hence, the time complexity is doubled, but still in $O(\log n)$.

If we do not have edge orientations or identifiers, the decision which of the nodes plays the part of the edge can, e.g., be made w.h.p. in a single round by exchanging random bit strings of size $O(\log n)$ between neighbors.

b) First, we 3-color the ring by means of Algorithm Six-2-Three (or its uniform variant from the first exercise sheet). Next, all nodes with color 0 join the dominating set and inform their neighbors. Then, all nodes with color 1 having no neighbor of color 0 join the set and inform their neighbors. Finally, still uncovered nodes with color 2 join the dominating set.

Obviously, the resulting set is a dominating set and the algorithm has a time complexity of $O(\log^* n)$. However, the constructed set is also a (maximal) independent set, as no two neighbors join. An independent set in a ring cannot have more than $n/2$ nodes, while a dominating set must contain at least $n/3$ nodes (each node covers itself and its two neighbors). In other words, the computed set is at most a factor of $3/2$ larger than any dominating set and hence also than a minimum dominating set.

c) Again we use one of the fast MIS algorithms to compute a maximal independent set I within $O(\log n)$ time. Observe that I is also a dominating set, due to the maximality of I; if there was a node that was not dominated, i.e., it is not in I and has no neighbor in I, this node would also be independent of I, which contradicts the assumption that I was a *maximal* independent set. Thus, we only need to show that I is at most C times larger than a minimum dominating set M.

To prove this, consider a node $v \in I$. Since M is a dominating set, there must be at least one node in $(N(v) \cup \{v\}) \cap M$, i.e., a node from the optimal solution is in v's neighborhood. For each $v \in I$, we count such a node. Because the graph is of bounded independence, no node $m \in M$ is counted more than C times, because there cannot be more than that many independent neighbors of m. Therefore, $|I| \leq C|M|$.