*CS-E4510 Distributed Algorithms / Jukka Suomela* *midterm exam, 23 October 2019*

**Instructions.** You must answer *all questions* in order to pass the exam. In all questions, it is sufficient to give a *brief, informal description of the algorithm*—you do not need to use the precise state-machine formalism, and you do not need to prove that your algorithm is correct. You are free to use any algorithms that were presented in the textbook as a subroutine; you do not need to repeat the details of those algorithms. If you cannot solve a problem entirely, please try to at least explain what you tried and what went wrong; *please make sure that you do not leave any question unanswered.*

**Question 1: PN.** Recall that a graph $G = (V, E)$ is a path graph if it is a tree and all nodes have degree at most 2. We define that a set of edges $M \subseteq E$ is *pleasant* if the following holds: for all nodes $v \in V$ of degree 2, there is exactly one edge $e \in M$ incident to $v$. Here is an example of a pleasant set of edges:

Give a deterministic distributed algorithm that solves the following problem in the PN model (any running time is fine):

- Graph family: path graphs.
- Local inputs: nothing.
- Local outputs: a pleasant set of edges (see above); each node has to indicate for each port whether the edge is in $M$ or not.

**Question 2: LOCAL.** An *orientation* of a graph $G = (V, E)$ assigns a direction to each edge $e \in E$. The *outdegree* of a node is the number of edges directed away from it. We define that a node is *happy* if it has outdegree equal to 1. We say that an orientation is *delightful* if for every edge $e \in E$, at least one of the endpoints is happy (i.e., the happy nodes form a vertex cover). Here is an example of a delightful orientation (2nd, 4th, and 5th node are happy):

Give a deterministic distributed algorithm that solves the following problem in time $O(1)$ in the LOCAL model:

- Graph family: path graphs.
- Local inputs: each node gets as input its own unique identifier.
- Local outputs: a delightful orientation of edges (see above); each node has to indicate for each port whether the edge is oriented away from the node or towards it.

You can assume that the unique identifiers are numbers from the set $\{1, 2, \ldots, n^2\}$.

**Question 3: CONGEST.** Give a deterministic distributed algorithm that solves the following problem in time $O(n)$ in the CONGEST model:

- Graph family: connected graphs.
- Local inputs: each node gets as input its own unique identifier.
- Local outputs: all nodes output "yes" if there exists a cycle in the input graph, and otherwise all nodes output "no".

You can assume that the unique identifiers are numbers from the set $\{1, 2, \ldots, n^2\}$.