

Instructions. There are three questions; please *try to answer something in each of them*. If you cannot solve a problem entirely, please try to solve at least some special case or simpler version of the problem (see the hints for suggestions), or failing that, please at least explain what you tried and what went wrong. Do not spend too much time with one problem; the problems are not listed in order of difficulty and they do not depend on each other.

You are free to look at any source material (this includes lecture notes, textbooks, and anything you can find with Google), but you are not allowed to collaborate with anyone else or ask for anyone's help (this includes collaboration with other students and asking for help in online forums). You are free to use any results from the lecture notes directly without repeating the details.

In each questions, it is sufficient to give a *brief, informal description* of the algorithm, and a *brief, informal explanation* of why it solves the problem correctly. You do not need to specify algorithms in the state-machine formalism, or give a complete proof of correctness. It is enough that a friendly, cooperative reader can understand your idea correctly and see why it makes sense. As usual, n denotes the number of nodes in the input graph.

Question 1: PN. 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: label the *edges* of the graph with *integers* such that
 1. *adjacent* edges have different labels and
 2. the total sum of all labels is 0.

Hints: If you have a path with 4 nodes and 3 edges, then examples of valid labelings include $0, +1, -1$ and $-123, +246, -123$. Please note that we did not promise that the path is directed—but if you cannot solve the general case, please try to solve at least the case of a directed path (i.e., a path where for each internal node there would be exactly one successor and one predecessor). Alternatively, you can try to solve at least the case of paths with an odd number of edges or the case of paths with an even number of edges.

Question 2: LOCAL. Give a deterministic distributed algorithm that solves the following problem in the LOCAL model in $o(n)$ rounds:

- Graph family: *cycle graphs*.
- Local inputs: each node gets as input its own unique identifier and the value of n .
- Local outputs: label the *nodes* of the graph with labels from the set $\{-1, 0, +1\}$ such that
 1. *adjacent* nodes have different labels and
 2. the total sum of all labels is 0.

Hints: Please note that the running time has to be $o(n)$, this is little- o , not big- O . So for example $\log n$ or $\log^* n$ is good. Also note that we did not promise that the cycle is directed—but if you cannot solve the general case, please try to solve at least the case of a directed cycle. If you cannot design a deterministic algorithm, try to design a randomized algorithm. In general, it may help to first think about the case of large values of n and then solve the case of a small n by brute force.

Question 3: CONGEST. Give a deterministic distributed algorithm that solves the following problem in the CONGEST model in $O(\text{diam}(G))$ rounds:

- Graph family: *connected graphs*.
- Local inputs: each node gets as input its own unique identifier.
- Local outputs: label the *nodes* of the graph with *integers* such that
 1. *all* nodes have different labels and
 2. the total sum of all labels is 0.

Hints: If you cannot solve this problem in diameter time, try to at least design an algorithm that solves it in $O(n)$ rounds.