CS-E4510 Distributed Algorithms / Jukka Suomela
exam, 8 December 2022

Material. You can bring one A4 paper (two-sided), with any content you want. No other material or equipment is allowed in the exam.

* * *

Definitions. In the *happy numbering* problem the task is to label each node v with a positive natural number $f(v) \in \{1, 2, \dots\}$ such that the following holds: if u is a node of degree at least 2 and s and t are two distinct neighbors of u, then

$$f(u) \neq \frac{f(s) + f(t)}{2}.$$

For example, if we have a path graph with 5 nodes, each of these is a happy numbering:

$$(1,1,2,1,1), \quad (3,2,3,2,3), \quad (2,3,3,5,9).$$

On the other hand, these are not happy numberings:

$$(1,1,1,1,1), \quad (1,2,3,4,5), \quad (2,5,8,1,2).$$

* * *

Questions. What can you say about the happy numbering problem in the models that we have studied in our course? Try to answer at least the following questions:

1. Can it be solved in the ***deterministic PN model*** in ***cycle graph***? How fast?
2. Can it be solved in the ***deterministic PN model*** in ***path graphs***? How fast?
3. Can it be solved in the ***deterministic LOCAL model*** in ***path graphs***? How fast?

* * *

Instructions. Whenever the answer is "no", please ***prove*** it. Whenever the answer is "yes", please give an ***algorithm*** that solves the problem, explain why it works correctly, analyze its running time, and if possible, ***prove*** that no algorithm can solve the problem (asymptotically) faster.

Your mathematical proofs and algorithm descriptions can be short and informal. It is enough that a friendly, cooperative reader can understand your idea correctly and see why it makes sense. You are free to refer to algorithms and results that we discussed in the lectures, course material, and exercises; there is no need to repeat any of their details.

If you cannot answer any of the above questions, please try to say ***anything*** meaningful (positive or negative) about the happy numbering problem in any of the models that we have studied in the course, for any graph family.