PS Algorithms for distributed systems

Homework Sheet 1

https://avs.cs.sbg.ac.at/

WS 2024/25

Submit until 02-12-2024 via email.

Homework 1.1 (3 points)

In the lecture we assumed that given a synchronous, uniform ring with unique identifiers, for the Clockwise algorithm, each node has a designated clockwise and counterclockwise port. Prove that even without this assumption, we can still appoint a leader in $O(n)$ rounds using $O(n^2)$ messages.

Homework 1.2 (3 points)

Given a synchronous, uniform ring with n nodes where each node has a unique identifier denoted by $v_1, ..., v_n$, let $\mathcal{A}$ be any deterministic algorithm that finds a unique leader. Prove that $\mathcal{A}$ needs at least $\frac{n}{4}$ rounds and $\Omega(n)$ messages. For simplicity, you may assume that $\frac{n}{4}$ is an integer.

Homework 1.3 (3 points)

In the lecture we have seen the Radius Growth algorithm for leader election by Hirschberg and Sinclair. In this algorithm, the radius grows by a factor 2 in each phase. Investigate what happens to the number of rounds and message complexity when the radius grows by a factor $k \geq 2$. Prove your bounds.

Homework 1.4 (6 points)

Write a program that elects a leader on an unweighted network in the CONGEST model (i.e. the nodes of the network have a unique ID and the communication via one edge per node per round has a bandwidth limitation of $O(\log(n)$ bits) in $O(D)$ rounds using the Pregel API of the GraphX component of the Apache Spark framework. You can test your code on a self-generated graph created using the graph generator within GraphX. The output shall print the elected leader, BFS tree structure and acknowledgements. Submit your code as a single file via email along with your other solutions.