Distributed Algorithms 2022/2023
(lab exercise)

Leader election

Task 1 — Implement a simulator that allows you to test the leader election algorithm presented during the lecture for known number of nodes n and for a known upper bound u on the number of nodes n. You can use any programming language.

Task 2 — Let the random variable L denote the number of slots from the start of the algorithm until the leader is elected. Use the simulator from the previous task to draw the empirical distribution (histogram) of the random variable L for both considered scenarios. For a scenario with a known constraint u consider three cases: $n = 2$, $n = u/2$, $n = u$. Justify the results. (10p)

Task 3 — For a scenario with a known number of n nodes, use the simulator to estimate $\mathbb{E}[L]$ and $\mathbb{V}ar[L]$. Verify that the results are consistent with the theoretical results. (10p)

Task 4 — Consider a scenario with a known constraint u. Let $S_{L,n}$ be the event that in one round of the algorithm of length $L = \lceil \log_2 u \rceil$ a leader was elected if there are n nodes in the system. Suggest a suitable experiment and use a simulator to confirm the correctness of the theorem from the lecture that $Pr[S_{L,n}] \geq \lambda \approx 0.579$. (10p)

Analysis of data streams: counting problem

MinCount

Task 5 — Implement the MinCount$(k, h, \mathcal{M})$ algorithm and test it:

a) Consider the multisets $\mathcal{M}_n = (S_n, m)$ such that $|S_n| = n$ for $n = 1, 2, \ldots, 10^4$ and all sets S_n are disjoint. Does changing the m function affect the value of the $\hat{n}$ estimation obtained in the algorithm?

b) For $k = 2, 3, 10, 100, 400$ and multisets from point a) draw a graph with n on the horizontal axis and $\hat{n}/n$ on the vertical axis.

c) Experimentally adjust the value of k so that there is 95% probability that $|\frac{\hat{n}}{n} - 1| < 10\%$.

(10p)

Task 6 — Test the MinCount$(k, h, \mathcal{M})$ algorithm for different hash functions $h : S \to \{0, 1\}^B$ and different values of parameter B. Try to find or propose a hash function h for which the accuracy of the algorithm is significantly worse than predicated by the analysis. Explain what causes this loss of accuracy. What else can matter besides the value of the B parameter? (10p)

Task 7 — Your task is to compare the theoretical concentration results for the $\hat{n}$ estimator used in the MinCount$(k, h, \mathcal{M})$ algorithm obtained by **a)** Chebyshev's inequality and **b)** Chernoff's inequality, with simulation results.

Namely, for $n = 1, 2, \ldots 10^4$, $k = 400$ and $\alpha = 5\%, 1\%, 0.5\%$ plot values of $\hat{n}/n$ obtained in simulation and values $1 - \delta$ i $1 + \delta$ such that

$$Pr\left[1 - \delta < \frac{\hat{n}}{n} < 1 + \delta\right] > 1 - \alpha\,. \quad (10p)$$

HyperLogLog

Task 8 — Implement HyperLogLog with corrections and test it for different values of the parameter m (number of registers) and different hash functions - create plots analogous to those from Task 5. Compare the estimation accuracy of the MinCount and HyperLogLog algorithms when both have the same amount of memory available (you can assume that HyperLogLog needs 5 bits per register and MinCount needs 32 bits per stored hash).

Blockchain

Task 9 — Let $0 < q < 1/2$ represent the probability of the adversary mining the next block and let this probability correspond to the fraction of computational power he/she possess. Let n represent the number of confirmations (mined blocks) needed to consider a transaction as confirmed. Let $P(n, q)$ represent the probability that an adversary with power q will mine a chain of blocks equal to or longer than the one mined by honest users at the moment when they have attached to the block containing the considered transaction n blocks, or ever after.

a) Compare the formulas for $P(n, q)$ obtained by Nakamoto and Grunspan. In particular:
   - set $n = 1, 3, 6, 12, 24, 48$ and plot $P(n, q)$ depending on the value of q,
   - set the acceptable probability of adversary success $P(n, q) = 0.1\%, 1\%, 10\%$ and draw graphs showing how to choose the value of n depending on the value of q.

b) Implement a „double spending attack" simulator, which will enable experimental approximation of the probability $P(n, q)$ depending on the values of n and q.
   Hint: design the experiment and repeat it multiple times (Monte Carlo method).
   Carefully and thoroughly describe the idea and code of the simulator.

c) Compare the simulator results to the analytical results (graphs). If there are discrepancies, try to explain them.

All the information you need was/will be given during the lecture. Additionally some notes are available in polish. (20p)

Self-stabilization

Task 10 — Implement the simulator of Dijkstra's Mutual Exclusion algorithm (page 17). For a given n denoting the number of processes in the ring, verify that starting from any initial configuration, the algorithm will reach a legal configuration. If it is possible to move to several possible configurations from a certain configuration, depending on which process makes the first step, each execution should be verified. What is the maximum number of steps to reach a legal configuration for a given n? For which values of n can you get an answer in a reasonable time? For this task, you can receive $3 \times$ N , points, where N is the largest value of n for which you can verify the algorithm.

Task 11 — Consider a graph $G = (V, E)$. We call two vertices $v, w \in V$ independent if $v, w \notin E$. We call a subset $S \subseteq V$ of vertices independent if all its elements are pairwise independent. Based on the Maximal Matching algorithm given in the lecture, design, implement and test a self-stabilizing algorithm that finds the maximal independent set (see Maximal Independent Set) in a connected undirected graph. Provide a convincing justification for the correctness of the algorithm (formal proof - exercise 34).
Algorithms for finding a maximal independent set have many applications. You might, for instance, think about the frequency assignment problem in wireless networks. (10p)