University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, G. Schmid

Algorithm Theory
Exercise Sheet 12

Due: Wednesday, 1st of Febuary, 2023, 11:59 pm

Exercise 1: Making Communication Possible *(10 Points)*

We consider an n player game, where all players use one shared communication channel. The goal is to successfully send any message, however if two players try to communicate at the same time we get a collision and nothing happens (picture a radio frequency).
The game works like this: time is measured in discrete rounds, where in each round each player either sends a message or does not. If more than 1 player tries to send a message we get a collision and all players will know that a collision occured (they don't know how many messages were sent). If no message was sent by any player, then also all players get the information that nobody sent that round. If exactly one player sent in a round, the message gets through and the game is won. No player knows how many players are in the game and all players follow exactly the same strategy.

(a) Argue that without any additional input no deterministic strategy can work! *(2 Points)*

Since a deterministic strategy is pointless we instead try a randomised strategy. We want to analyse the simple strategy, in which all players will send with the same probability p.

(b) Prove that for $p = \frac{1}{n}$ the success probability is maximised and that when using $p = \frac{1}{n}$, in expectation we only need $O(1)$ rounds to win the game. *(5 Points)*

Since our players don't know n we cannot simply send with $p = \frac{1}{n}$, so instead we try to guess what n is, for that we start with our guess $g = 2$ and send with $p = \frac{1}{g}$. If there is a collision, we conclude that our estimate is too low and update $g \leftarrow g \cdot 2$ and if we don't get any message, we conclude that our estimate is too high and update $g \leftarrow g \cdot \frac{1}{2}$.
Sadly the complete proof is a bit too complex for this sheet, but the intuition as to why this works should be clear from the following fact:

(c) Prove that for any two constants $0 < c < c'$, if we have $p \in [\frac{c}{n}, \frac{c'}{n}]$ the expected time until we win is still $O(1)$. *(3 Points)*

That is, if we are in the general vicinity of $\frac{1}{n}$ we still will win very fast.

Exercise 2: i-th smallest Element *(10 Points)*

We are given a set A of n distinct numbers, we are interested in an algorithm that, given some $i \in \mathbb{N}$, returns the element $x \in A$ such that there are exactly i elements smaller than x in A. A trivial solution would be to just sort the array in $O(n \log n)$ and then return the element at position i. However that involves a lot of unnecessary work. Give a randomised algorithm that makes $O(n)$ comparisons in expectation.