Dr. Alexander Munteanu
Dennis Köhn, Dr. Vibha Sahlot

Winter term 2024/2025

3. Home Exercises

Efficient Algorithms

Ex. 1 *(4 points)*

A fair coin gets tossed $n + \log_2(n)$ many times, where n is a power of two. This creates a sequence $x_1, x_2, ...x_{n+\log_2(n)}$ with $x_i \in \{H, T\}$, where H represents heads and T represents tails.

We call $x_i, ..., x_{i+l-1}$ an *l-sequence* if $x_i = x_{i+1} = ... = x_{i+l-1}$ (all heads or tails). What is the expected number of l-sequences for $l = 1 + \log_2 n$?

Ex. 2 *(4 + 2 + 2) Points*

Consider the k-**Median** problem introduced in the lecture. As before, the input is a metric space (P, d). Let $C = \{c_1, \ldots, c_k\}$ be a set of optimal centers and let $\mathrm{OPT} = \sum_{p \in P} d(p, C)$.

a) Develop a randomized algorithm that computes a weighted set S, such that

$$\mathbb{E}\left(\sum_{s \in S} w(s) \cdot d(s, C)\right) = \mathrm{OPT},$$

where $w(s)$ denotes the weight of point s. The running time of the algorithm should be $O(n)$ and the size of S should be $O(1)$.

b) Analyze the running time of your algorithm.

c) Prove the correctness of your algorithm.

Ex. 3 *(4 + 2 + 2 Points)*

Let $a_1, \ldots, a_n$ be a sequence of n distinct natural numbers. The goal of this exercise is to compute a random permutation of this sequence of numbers. The naive way to do this would be to produce a list of n priorities and sort the numbers based on their priority. However, sorting takes $O(n \log n)$ time and we want to do this in linear time.

a) Design an algorithm that outputs an uniformly distributed random permutation of the given sequence in running time $O(n)$.

You may assume that you have access to a function $RANDOM(k)$ that outputs a uniformly distributed random number from $\{1, \ldots, k\}$ in time $O(1)$.

b) Analyze the running time of your algorithm.

c) Prove the correctness of your algorithm.

Hint: prove that any possible permutation is output with probability $1/n!$.