Assignment 3

Solve the following three exercises. Each exercise is worth five points. Solutions must be submitted by 18.4.2022. Use the link on e-ucilnica to submit your solutions. Solution must be submitted in pdf format.

To pass the course, you must score at least 50% on each assignment (not each exercise). All assignments together are worth 100 points.

Exercise 1: Amortization

You are working on an algorithm, which is adding rows and columns to a matrix. Each call to an Add() function costs $i + c$ where i is the i-th call of the function and c is some constant. Every call where $i = k^2$ for some $k \in \mathbb{N}$ costs i^2. Mathematically the cost function is :

$$c_i = \begin{cases} i + c & ; i \neq k^2, k \in \mathbb{N} \\ i^2 & ; i = k^2, k \in \mathbb{N}. \end{cases}$$

What is the amortized cost of this function?

Exercise 2: Amortization

You are working on a dynamic table which only support inserts. Instead of doubling table size when it is full you decide to increase it for 10% only. Is amortized cost of insert still constant? Prove using potential method.

Exercise 3: Approximation

Suppose you are given a *symmetric* 4-SAT formula, which is described with 4-CNF formula F with n clauses, each clause consisting of 4 literals. For example:

$$F_e = (x_1 \vee x_2 \vee \neg x_4 \vee x_5) \wedge (x_4 \vee \neg x_2 \vee \neg x_1 \vee x_3) \wedge (\neg x_3 \vee x_2 \vee \neg x_5 \vee x_1)$$

In 4-SAT we accept each clause if the clause evaluates to 1. In *symmetric* 4-SAT we accept a clause if it evaluates to 1 and a symmetric clause in which we negate each literal also evaluates to 1. For example we accept the first clause of F_e if the following holds:

$$(x_1 \vee x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4 \vee \neg x_5) == 1$$

Or in other words *symmetric* 4-SAT accepts each clause if it has one literal that assigns to 0 and one that assigns to 1.

We know that *symmetric* MAX 4-SAT in which we try to satisfy as many clauses as possible is a NP-complete problem. We make a simple approximation algorithm. Let us set each variable to 0 with probability 0.5 and to 1 with probability 0.5.

Find the approximation factor for this algorithm.

Note: In 4-CNF each clause can not have the same literal twice or have a variable x_i and its negation $\neg x_i$.