**Analysis of Algorithms and Heuristic Problem Solving,** 2021/22, 08 June 2022, Written exam**.**

All questions count equally. Literature, electronic and communication devices are not allowed. It is allowed to use 1 sheet of A4 format paper. You can write your answers in either English or Slovene. Duration: 90 minutes.
Students who wish to look into the written exam results can do so on Monday, 13 June 2022, at 12:00 in the room of Prof Robnik Šikonja (2nd floor, room 2.06).

1. Find the solution to the recurrence:
   $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{6}\right) + T\left(\frac{n}{12}\right) + n$

2. Consider a very simple online auction system that works as follows. There are *n* bidding agents; agent *i* has a bid $b_i$, which is a positive natural number. We will assume that all bids $b_i$, are distinct from one another. The bidding agents appear in an order chosen uniformly at random, each proposes its bid $b_i$ in turn, and at all times the system maintains a variable $b^*$ equal to the highest bid seen so far. Initially, $b^*$ is set to 0.
   What is the expected number of times that $b^*$ is updated when this process is executed, as a function of the parameters in the problem?
   Example. Suppose $b_1$ = 20, $b_2$ = 25, and $b_3$ = 10, and the bidders arrive in the order 1, 3, 2. Then $b^*$ is updated for bidders 1 and 2, but not for 3.

3. Given a set of *m* linear inequalities on *n* variables $x_1, x_2, \dots x_n$, the linear inequality feasibility problem asks whether there is a setting of the variables that simultaneously satisfies each of the inequalities. Show that if we have an algorithm for linear programming, we can use it to solve a linear-inequality feasibility problem. The number of variables and constraints that you use in the linear-programming problem should be polynomial in *n* and *m*.

4. You are given a task to solve the facility assignment problem defined as follows. There is a set *U* of users (defined with locations) that need access to a service, and a set of possible server locations *S*. For each site $s \in S$, there is a fee $f_s \geq 0$ for placing a server at that location. Users $u \in U$ can be served from multiple sites, with associated cost $c_{us}$ for serving user *u* from site *s*. If cost $c_{us}$ is high, we will avoid serving user u from site s; in this way we can promote serving users from nearby sites.
   For sets *U* and *S*, and cost functions *f* and *c*, you have to select a subset $A \subseteq S$ at which to place servers and assign each user to the active server where it is cheapest to be served.
   a) Formally define an objective function that minimizes the total cost of placing the servers and serving the users.
   b) Propose a data structure to represent the problem and describe a function that generates neighborhood to solve this task with local optimization.