Computational Complexity and Heuristic Programming, 2016/17
Written exam, 16 February 2017

All questions count equally. Literature, electronic and communication devices are not allowed. It is allowed to use up to 2 sheets of A4 format paper with handwritten notes. You can write your answers in Slovene. Duration: 90 minutes.
Oral exam for students who wish to improve their grade and have achieved at least 50% of points in written exam, will take place on Monday, 20 February 2017 at 13:00, in the office of Prof Robnik Šikonja (2nd floor, room 2.06).

1) a) Find a solution to the following recurrence.

$$T(n) = 4T(n-1) - 3T(n-2) + 1$$

b) Given the following recurrence

$$T(n) = T(n-1) * T(n-1)$$
$$T(1) = 2.$$

prove using induction that $T(n) = \Omega(2^n)$

2. Assume you are creating an array data structure that has a fixed size of *n*. You want to backup this array after every so many insertion operations. Unfortunately, the backup operation is quite expensive; it takes *n* time units to do the backup. Insertions without a backup just take 1 time unit. You backup the array after every *n* insertions.
Prove that you can do backups in O(1) amortized time. Use the potential method for your proof.

3. In a given array *A* of length *n* we store unsorted integers. Suppose you are searching for an integer *x* in *A* using the following procedure. Select a random index *i*. If A[*i*] = *x*, return *i*, otherwise select another index *i*. Repeat the procedure until you find the element *x* or all elements have been scanned. Note that index *i* is always selected from all available elements and repetition is possible. Find the expected number of checked elements if the array *A* contains exactly one element equal to *x*. Justify your answer.

4. A graph bandwidth problem is to label the *n* vertices v_i of a graph G=(V,E) with distinct integers $f(v_i)$ so that the quantity $\max\{| f(v_i) - f(v_j) | : (v_i,v_j) \in E\}$ is minimized. The problem may be visualized as placing the vertices of a graph at distinct integer points along the x-axis so that the length of the longest edge is minimized.
Propose a neighbourhood structure for local optimization approach to this problem and propose features which could be used in penalization with guided local search. Suggest contents of tabu lists which could be useful in tabu search. Justify your suggestions.