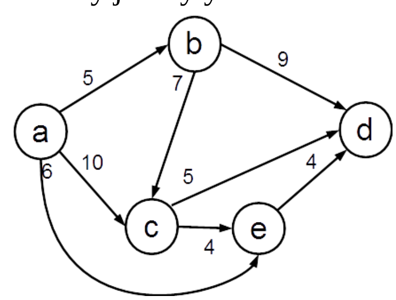
Computational Complexity and Heuristic Programming, 2016/17
Written exam, 23 August 2017

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, 28 August 2017 at 13:00, in the office of Prof Robnik Šikonja (2nd floor, room 2.06).

1. Find and solve a recurrence relation to explain the result of the following procedure.
   i) Choose any two integers and write them one after another in ascending order.
   ii) Form a third number by subtracting the first number from 5/2 times the second number. Form a fourth number by subtracting the second from 5/2 times the third, etc. until you have a sequence of n numbers.
   If you divide the n-th number by the (n-1)-th, the value you get should be very close to 2.

   To explain the result write down a recurrence relation f(n) for the n-th number in the sequence. Get the general solution to this recurrence relation using annihilators. Next, figure out what is a good approximation of f(n) to this solution for large n.

2. Consider using a simple linked list as a dictionary. Assume the client will never provide duplicate elements, so you can just insert elements at the beginning of the list. Now assume the peculiar situation that the client may perform any number of insert operations but will only ever perform at most one lookup operation.
   a. What is the worst-case running-time of the operations performed on this data structure under the assumptions above? Briefly justify your answer.
   b. What is the worst-case amortized running-time of the operations performed on this data structure under the assumptions above? Briefly justify your answer.



3. Write a linear program which finds a shortest path from node *a* to node *d* in the given graph.

4. A clique cover of a given undirected graph is a partition of the vertices of the graph into cliques i.e. subsets of vertices within which every two vertices are adjacent. A minimum clique cover is a clique cover that uses as few cliques as possible. We want to find the minimum k for which a clique cover exists.
   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.