Data structures and algorithms
(2019/20)
Written exam 30. prosinca 2020

This written exam must be taken individually. Any and all literature may be used while taking this test. In your answers be precise, and: (i) answer the questions *as they were asked*; and (ii) answer *all* tasks – if you will be answering to all tasks you might get bonus points.

Time: 90 minutes.

We wish you a lot of success - veliko uspeha!

NALOGA	TOČK	OD TOČK	NALOGA	TOČK	OD TOČK
1			3		
2			4		

IME IN PRIIMEK: _____

ŠTUDENTSKA ŠTEVILKA: _____

DATUM: _____

PODPIS: _____

1. naloga: Basics. Bubblesort is one of the oldest sorting algorithms:

```
int[] Sort (A):
  for i = len(A)-1, ... 1:
    for j = 0 ... i-1:
      if A[j] >= A[j+1]:
        tmp = A[j+1]; A[j+1] = A[j]; A[j] = tmp
  return A
```

In general, let $|\mathtt{A}| = n$.

QUESTIONS:

A) (i.) What precisely does the inner `for` loop on `j`? (ii.) Prove the correctness of your claim.

   HINT: In the proof, use induction and loop invariant. Note that the inner loop depends on `i` and not on n.

B) (i.) What is the minimal number of *comparisons* in the function `Sort(A)`? Show in which case this happens. (ii.) And what is the maximal number of comparisons in the function `Sort(A)`? In which case this happens?

C) (i.) Is the sorting in the above bubblesort code stable or not? (ii.) Prove your claim. (iii.) If it is not stable, can you modify the code so that the sorting will become stable? Justify your solution.

2. naloga: Balanced trees.

QUESTIONS:

A) We use single and double rotations to balance AVL trees. (i.) Draw an example of a single rotation. In figure, include also the heights of subtrees and show how they change. (ii.) Draw an example of a double rotation. In figure, include also the heights of subtrees and show how they change.

B) The balance happens at insertions and deletions in AVL trees. (i.) When inserting, what is the maximal number of rotations that can happen to balance the tree? Justify your answer. (ii.) When deleting, what is the maximal number of rotations that can happen to balance the tree? Justify your answer.

C) Our friend Peter Puzzle heard that the data structure dictionary can be augmented to support the query `Num(a, b)`, that returns the number of those elements in the dictionary that are contained in the interval (a, b). But he does not know if the augmentation can be done efficiently for every implementation of the dictionary. (i.) List three implementations we have mentioned in

the lectures and require $O(n)$ time complexity for the described operation, where n is the number of elements in the dictionary. (ii.) Justify your answer.

3. naloga: Tries. Peter Puzzle found the following text

$$t = GACCGAGTAGAG \ . \tag{1}$$

QUESTIONS:

A) (i.) From the text in (1) build a suffix tree. (ii.) Which pattern of length 2 is the most often repeated? Which pattern of length 3 is the most often repeated?

B) (i.) Describe a data structure that permits to find which pattern of length k is the most often repeated in a text of length n. (ii.) Describe a procedure for finding which pattern of length k is the most often repeated in a text of length n. (iii.) What is the space complexity of your structure? Justify your answer.

   HINT: A data structure design can be trie based and appropriately augmented.

C) What is the time complexity of your procedure? Justify your answer.

   HINT: In your solution, you may assume that a suffix tree is already built. The time complexity should depend on n and k.

4. naloga: Graph algorithms. Butale are really weird country. Even more strange are the mayors and their decisions. The latter is particularly interesting, namely the mayor Francot Turkavidel passed a decree that all roads in Butale must be one Butale mile long. Peter Puzzle immediately felt a business opportunity and came to offer an online service that calculates the shortest path between two places in Butale. Let n be the number of places in Butale and let m be the number of roads.

QUESTIONS:

A) (i.) Describe a *data* structure that Peter should use for efficient implementation of his service. (ii.) Write an algorithm that calculates the distance between the places a and b. (iii.) What is the time complexity of your algorithm? Justify your answer.

B) Peter decided to compute the shortest paths once and for all among all the pairs of places in Butale. (i.) Write down the Floyd-Warshal algorithm for finding the shortest paths between two arbitrary places but in such a way that it will be possible to reconstruct the path. (ii.) Can Peter compute it faster than using the Floyd-Warshal's algorithm? Justify the answer.

C) New mayor and new decisions. Francot's son Gregor Brezhlačnice slightly modified his father's decree by allowing that the roads can now be one or two Butale miles long. (i.) What shall Peter do so that he could still use the solution from A? Justify the answer. (ii.) How does the time complexity of the solution change? Justify the answer.