Data structures and algorithms
(2021/22)
Written exam February 4th, 2022

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: 105 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. task: Basics. We are given an array `A[1..n]` of integers. Peter wants to find the longest increasing contiguous subsequence in the array. For example, in the array

$$[92, 25, 67, 85, 79, 50, 83, 12, 14, 29, 46, 28]$$

this is the subsequence $[12, 14, 29, 46]$ with length 4.

A) (i) Write down an algorithm that in an array of length n finds the longest increasing contiguous subsequence. (ii) Show its correctness.

B) (i) What is the time complexity of your algorithm? Justify your answer. (ii) Could you find the subsequence faster? Justify your answer.

C) (i) Sort the following functions by ascending order:

$$n^{\frac{3}{2}},\ \frac{n}{\log n^2},\ \frac{n}{\log n},\ n^{1-\epsilon}.$$

(ii) Justify your answer.

2. task: In Butale, they have a very long fence and since it is already old, they decided to repaint it. To make it more fun, each resident of Butale can decide which part of the fence (s)he would like to repaint. They asked Peter Puzzle to prepare the function `Paint(a, b)` that every resident of Butale would call and thereby announce his/her wish. For example, Luka wanted to repaint the fence from 123m to 132m (inclusive) and so he called the function `Paint(123, 132)`. For *each question*, (i.) suggest a data structure and implementations of the function `Paint()` and of the function required by an individual question; and (ii.) for both functions provide the time complexities and justify your answer.

QUESTIONS:

A) First, the residents of Butale wanted to know which most left and which most right meter should be painted. In addition to `Paint()`, another function is `LeftRight()`, which returns the required answer.

B) The next function that the residents of Butale wanted Peter to implement was `Painted(a, b)`, which returns TRUE, if the fence is completely painted between the meters `a` and `b` (inclusive).

C) Clearly, because the desires of the residents of Butale were uncoordinated, there can be more of them painting the same meter. This time the mayor of Butale is interested how many times will be repainted each meter of fence between `a` and `b` (inclusive). In other words, the function `AtLeast(a, b)`

finds in the given section the meter which would be painted by the fewest residents and returns their number. For example, after calls `Paint(10, 14)` and `Paint(8, 12)`, the function `AtLeat(10, 13)` returns 1, because at least one resident of Butale wants to paint every meter between `10` and `13` (inclusive).

HINT: The better your solution, the more points you get.

3. task: In Butale country, they are renovating a power plant and they have to bring a very wide spare part. Suppose that their road system is represented by a graph $G(V, E)$, where V is the set of n nodes and E the set of m edges between nodes where the edge is given as $(u, v; w)$ and w is the smallest road width between u and v. The problem which needs to be solved on the graph is to find the widest path between nodes s and t.

QUESTIONS:

A) Peter Puzzle suggested the following non-deterministic algorithm which returns as a certificate the widest path between s and t:

```
def Widest(s, t):
  path= "s"
  repeat
    u= Nondeterministically Choose Next vertex
    path= Extend(path, u)
  unitil u = t
  return path
```

(i.) In what way does the certificate prove that the path found is the widest? Does the certificate convince us?

B) Write down a deterministic algorithm that finds the widest path between s and t.

C) (i.) What is the space and time complexity of your algorithm? Justify your answer. (ii.) Compare the time complexity of your and Peter's solution. For the latter, consider a possible verification of the certificate.

4. task: Who finds the longer one? We are playing the next game. We have a cyclic array of integers $A = [a_0, a_1, \ldots, a_{n-2}, a_{n-1}]$, where a_{n-1} is again followed by a_0 and so on.

In the cyclic array A we look for descending sequences. For example, if we have the array $A = [22, 65, 57, 63, 92, 61, 45]$, the following are examples of descending sequences $[65, 57]$ and $[92, 63, 45, 22]$, where the latter exploits the cyclic

structure of the array A. The problem we need to solve is: *in a given cyclic array A of length n find the longest descending sequences (LDS).*

QUESTIONS:

A) Let $a_m = \min_{i=0..n-1} a_i$. Is a_m always in LDS? Justify your answer.

B) Show that the greedy method in developing an algorithm that finds LDS does not work.

C) (i.) Write down an algorithm that finds LDS. (ii.) What is the space and time complexity of your algorithm? Justify your answer.