Data Structures and Algorithms
(2017/18)
Exam June 15, 2018

This test 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: Dictionary. We know several dictionary implementations, but they all support basic operations `insert(x)`, `delete(x)` and `find(x)`.

VPRAŠANJA:

A) Suppose that Peter Zmeda wants to add the operation `range(x, y)`, which returns a list of all elements z in the dictionary such that $x \leq z \leq y$. Help him and write down an implementation of `range()` for the dictionary represented by AVL tree, skip list and hash table with open addressing.

   NAMIG: The more precise your implementation, the more points you will earn – for example, beware of the data structures used etc.

B) For each of the three implementations of the function `range()` from the first question, evaluate its time complexity and justify the answers.

C) Suppose we add an additional request; namely, the returned list of elements should be sorted by value. (i) Are there any changes in the time complexity of the implemented functions? Justify your answer. (ii) Which implementation of the dictionary would you choose if you have to expand it with the function `range()`? Justify your answer.

2. naloga: Tries. Peter Zmeda also works on bioinformatics, where his alphabet consists of four characters $\Sigma = \{\text{A}, \text{C}, \text{G}, \text{T}\}$. Frequently used data structure in bioinformatics is a suffix tree, which is in fact a trie, which contains all suffixes of a string. For example, the string $s =$ TTATGTA has eighth suffixes, where the first suffix is the string s itself, the next one is TATGTA, then ATGTA, and so on till A and the empty string.

VPRAŠANJA:

A) (i) Draw a trie containing all suffixes of the string s. In leaves, should also be stored indices representing the starting positions of the suffixes in the text. For example, `atgta` starts at the 3$^{\text{rd}}$ position. (ii) Change trie to the PATRICIA tree – we call this tree a suffix tree.

B) (i) Write down an algorithm that constructs a suffix tree from a given string of length n. (ii) What is its time complexity? Justify the answer. (iii) What do you think is the time complexity of the fastest possible algorithm? Justify answer.

   NAMIG: Consider how many nodes a suffix tree contains and how this affects the time complexity.

C) Suffix trees allow a quick search for the pattern of length m in a string in time $O(m)$. (i.) Describe the search process. (ii.) Now, we will expand the search function to `find(p1, p2, k)`, which finds a pattern of length k in a text that begins with `p1` and terminates with `p2`. For example, in this way we can search for the patten TAT.TA (there are arbitrary letters at the dot position) by calling the function `find(TAT, TA, 6)`. In the string s the pattern is placed at the second position. Suggest an effective data structure and an algorithm for such a search and justify your answer.

   NAMIG: Only a suffix tree will not suffice. Think how can you use information stored in the leaves of the tree. The answer is surprisingly short.

3. naloga: Graphs. Relationships between people can be modelled using graphs. In Butale, there are particularly proud to have two relationships: *know* in *best friend* (or *friend* for short). The relation $A\,\mathrm{know}\,B$ means that person A knows B, but the converse is not necessarily true. On the other hand, the relation $A\,\mathrm{friend}\,B$ means that B is the best friend of A and that A and B mutually know each other. Of course, one can have only one best friend and the latter must exist.

In Butale, there have been offered a new service *ButBum* (*Butale's Album*), in which each person enters his best friend.

VPRAŠANJA:

A) To implement ButBum service, they contracted with Peter Zmeda. They are expecting from him to support the following operations: `AddFriend(A, B)`, which adds the best friend B of A. Here, both A and B should exist. In addition, if A has already had the best friend C, C is not the best friend of A anymore. The second operation is `Friend(A)`, which returns the best friend of the person A. If a person A does not exist, or if (s)he exists and does not have the best friend, it returns NULL. (i) Suggest the data structure Peter should use for the implementation and give the implementation of both operations. Justify the correctness of your decision on a structure and correctness of your code. (ii) What is the time complexity of your operations? Justify the answer. (iii) Suppose that the relation *friend* is modelled by a graph. What is this graph? Justify your answer.

   NAMIG: We usually justify a decision by making it effective.

B) The work is done. Almost. People from Butale have come up with an idea to expand ButBum with an additional service; namely, `Invite(A, text)`, that sends the invitation on the birthday party of A with text `text`. Because they are social people, the invitation should receive the best friend of A, then the best friend of this best friend and so on. (i) Implement the new operation. (ii) What is the time complexity and justify the answer.

C) Successful entrepreneurship is an endless source of new ideas. Now, we would like to upgrade ButBum and add the third function; namely, support for a relationship *know*. (i) Suggest a data structure to support the new function and justify the answer. (ii) To startup the new operation, can ButBum use the data they already have with offering the operation *best friend*? Justify the answer.

NAMIG: The more precise your answer, the more points you will earn.

4. naloga: Optimization problems. Peter Zmeda has opened a translation company in Butale. He plans the work only for the current day so that in the morning he sends a request for translations and clients send their texts him back together with the information of how much they are ready to pay for each translation. Peter then reviews their texts and evaluates how much time he would need for each translation. In the end, he decides which texts he will accept to translate such that his profit is maximized.

VPRAŠANJA:

A) Today he has received seven texts that he has examined and found the next estimates on the time needed to translate a single text and the price a client is ready to pay:

text	1	2	3	4	5	6	7
time	13	19	7	7	5	5	3
price	35	24	6	20	11	20	15

Give advise to Peter, who has 24 hours for translations, which texts he should take in order to maximize the profit. Justify the answer.

B) Write down an algorithm that for n texts, where Peter for the i-th text needs t_i time to translate it and earns on it c_i, returns *which texts* should Peter accept to translate so that his profit is maximized. He has T time for translations.

NAMIG: In the above example, $n = 7$ and $T = 24h$.

C) Peter expanded the company and hired two other translators. Write a *non-deterministic* algorithm that distributes work to all three translators so that the total profit is at least C. Justify the correctness of the algorithm.

NAMIG: Consider what a certificate is.