Algoritmi in podatkovne strukture
Second Midterm (2016/17)

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

Time: 60 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: Peter Zmeda is about to do his homework. He needs to solve seven tasks and for each task he earns a number of points. Also, he estimates to spend the following amount of time for each task:

points	7	9	5	12	14	6	12
time (in h)	3	4	2	6	7	3	5

To complete his homework, Peter has 15 hours of left. Obviously he cannot complete all the tasks, so he needs to decide which tasks to take in order to maximize the number of points. This is an optimization problem.

VPRAŠANJA:

A) Suppose Peter earns partial points for partially completed tasks. This means, if he solves the half of the first task, he spends 1.5h and earns 3.5 points. (i) Which tasks should Peter solve and to what extent in order to earn as much points as possible? (ii) Justify the correctness of your approach.

B) Now suppose Peter will earn points only, if he completes a task (the principle of all or nothing). (i) Which tasks from the table above should Peter take now? (ii) Justify the correctness of your answer.

C) In general, what if there are n tasks to consider and we need to find an optimal solution acknowledging the all or nothing principle? Write down an algorithm.

2. naloga: Suppose we have two strings

$$t_1 = \texttt{05937299952727}$$
$$t_2 = \texttt{9952727369263848}$$

where each character is from the alphabet $\Sigma = \{\texttt{0},\texttt{1},\texttt{2},\texttt{3},\texttt{4},\texttt{5},\texttt{6},\texttt{7},\texttt{8},\texttt{9}\}$. Let p be a substring of t. For example `999` is a substring of t_1 and `27` is as well.

VPRAŠANJA:

A) The longest common substring of strings u and v is a substring for which there is no other longer substring present in both strings. Find the longest common substring of strings t_1 and t_2.

B) (i) Write down an algorithm for finding the longest common substring of two strings u and v, where $|u| = n_u$ and $|v| = n_v$. (ii) What is the time complexity of your algorithm? Justify your answer. (iii) Do you think it is possible to find the longest common substring faster? Justify your answer.

C) Let us turn our problem the other way around. We still have two strings u and v, where $|u| = n_u$ and $|v| = n_v$, but we obtain a substring p_{uv} from Peter for which he claims it is the longest common substring of u and v. (i) Write down the algorithm which checks, whether p_{uv} is truly the longest common substring. (ii) Evaluate and justify the time complexity of your algorithm. (iii) Can the time complexity of the algorithm be improved?

3. naloga: In sorting there is a known term *permutation vector* π – a vector of length n, where the element at position i is moved to position $\pi[i]$. The permutation vector can be presented as a directed graph. For example, the vector $\pi = [3, 0, 2, 4, 1, 5]$ is presented in Figure 1. There are three cycles in graph G

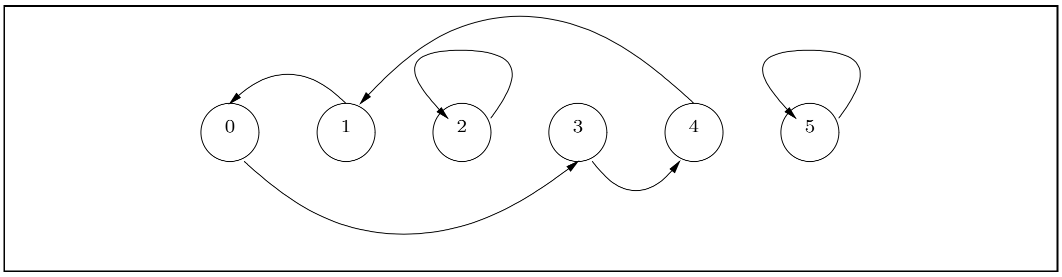

Figure 1: Graph $G(V, E)$ of the permutation vector π.

where the longest one is of length 4 ($0 \to 3 \to 4 \to 1 \to 0$), while the other two are of length 1.

VPRAŠANJA:

1. (i) If the length of the permutation vector is n, what are the cardinalities of sets V and E? Justify the answer. (ii) Write down an algorithm, which finds all cycles in the permutation vector. Justify its correctness. (iii) What is the time and space complexity of your algorithm? Justify your answer.

2. How would you parallelize this algorithm? Is there any other algorithm more convenient for parallelization?

   HINT: The better your parallelization is, the more points you will earn.

3. Let π_R and π_G be two permutation vectors[1] of length n. Each vector defines its graph $G_R(V_R, E_R)$ and $G_G(V_G, E_G)$, where $V_R = V_G$. We define the union of two graphs $G = G_R \cup G_G$, where $V = V_R = V_G$ and $E = E_R \cup E_G$. (i) Write down an algorithm, which finds in G such shortest cycle in which colors on edges alternate: red, green, red and so on. Justify the

[1] R stands for (*red*) and G (*green*).

correctness of your algorithm. (ii) What is the time and space complexity of your algorithm? Justify your answer. (iii) Write down an algorithm, which finds in G such longest cycle in which colors on edges alternate: red, green, red and so on. Justify the correctness of your algorithm.

4. naloga: One of the latest innovative algorithms is called `Bogosort`:

```
Bogosort($A$)
  REPEAT forever:
    **Randomly choose** permutation $\pi$ of length $n$
    IF array $\pi(A)$ is sorted THEN return $\pi(A)$
```

Let A be an array of n integers. For example, if:

- $A = (23, 44, 11, 15)$ and
- a random permutation $\pi = (1, 3, 2, 4)$,
- then $\pi(A) = (23, 11, 44, 15)$, and
- the array is not sorted.

VPRAŠANJA:

A) (i) What is the time complexity of `Bogosort` in the worst case? Justify your answer. (ii) What is the expected time complexity of `Bogosort`? Justify your answer.

B) Is `Bogosort` a kind of Las Vegas or Monte Carlo algorithm? Justify your answer.

C) (i) Design a parallel version of `Bogosort` running on p processors. We will evaluate the correctness of your algorithm and the achieved speedup. (ii) What is the time complexity of your solution? Justify your answer.