Data structures and algorithms (2022/23) Written exam February 15th, 2023

This 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! Veliko uspeha!

TASK	POINTS	OF POINTS	TASK	POINTS	OF POINTS
1			3		
2			4		

NAME AND SURNAME: ______________________

STUDENT ID: ______________________

DATE: ______________________

SIGNATURE: ______________________

Task 1. *Priority queues.* A structure is always a *min* structure. We have the following sequence of operations:

`I1, I2, I3, I4, I5, I6, I7, M, I8, DM, I9, DM, I10,`

where `M` represents returning the minimal element, `DM` deleting the minimal element and `Ix` inserting an element `x` into a priority queue.

QUESTIONS:

1. (i) Perform the above operations over an empty binary heap. Draw the structure after each operation. (ii) Now perform the above operations over an empty lazy binomial heap. Again, draw the structure after each operation.

2. Suppose we are given elements from 1 to n in a priority queue. (i) Where the element n in a binary heap is located? Justify your answer. (ii) Where can the element n in a (non-lazy) binomial heap be located? Justify your answer.

3. Now we have a binary heap that contains elements with values from 1 to $n = 2^k - 1$. What is the maximum possible value of a child of the root of the entire heap? Justify your answer.

Task 2. *Dictionary.* Suppose we have an array of integers a_i $(0 < i)$:

$$4\ 6\ 7\ 8\ 10\ 11\ 11\ 11\ 12\ 16\ 19\ 19\ 20\ 25\ 26\ . \tag{1}$$

QUESTIONS:

1. (i.) Insert the integers from the array (1) into a unary search tree and draw it. (ii.) Insert the integers from the array (1) into a ternary search tree and draw the final tree. The final tree should be as shallow as possible, but still allow searching. (iii.) How deep can such a ternary search tree with arbitrary n elements that can be repeated be? Justify the answer.

HINT: If we search for all occurrences of an element in the tree, this should not get us to traverse the entire tree. At the same time, each copy of the element should have its own node in the tree.

2. In general, we have an array of strictly increasing integers (this time the numbers are not repeated) a_i, $i = 1, 2, \ldots n$, $a_{i+1} > a_i$. (i.) Write an algorithm that builds as shallow as possible binary search tree from the integers a_i. (ii.) Justify the correctness of your algorithm. (iii.) What is the time complexity of your algorithm? Justify the answer.

$\boxed{\textbf{3.}}$ We have again an array of n non-decreasing numbers (this time the numbers repeat) a_i, $i = 1, 2, \ldots n$, $a_{i+1} \geq a_i$. (i.) Write an algorithm that builds as shallow as possible binary search tree from the integers a_i. (ii.) Justify the correctness of your algorithm. (iii.) What is the time complexity of your algorithm? Justify the answer.

Task 3. *Algorithm design.* Supoose we have an array of integers a_i $(0 < i)$ and the task is to find $a_i = i$. In the array (1) such an index i does not exists, while it exists in

$$5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ 5\ .$$

QUESTIONS:

$\boxed{\textbf{1.}}$ We have an array of non-descending integers a_i, $i = 1, 2, \ldots n$, $a_{i+1} \geq a_i$ (integers can be repeated). (i.) Write an efficient algorithm to determine whether $a_i = i$. (ii.) What is the time complexity of your algorithm? Justify the answer. (iii.) Show that your algorithm is correct, meaning that it returns *yes* if and only if $i = a_i$ actually exists.

$\boxed{\textbf{2.}}$ This time we have an array of strictly increasing integers a_i, $i = 1, 2, \ldots n$, $a_{i+1} \geq a_i$ (integers are not repeated). Again, (i.) write an efficient algorithm to determine whether $a_i = i$, (ii.) find out and justify the time complexity of your algorithm and (iii.) show that your algorithm is correct.

Task 4. *Graphs.* The mayor of Butale was invited by the mayor of New York to the celebration. At this the mayor of Butale visited the city and very much liked the network organization of roads and avenues in Manhattan as it has been laid down in year 1811 (see Fig. 1). The network can be described as a graph $G_M(V_M, E_M)$, where the nodes are intersections and between them roads (horizontal) and avenues (vertical) are edges. In this case, the road connections are considered to be 750 feet long and avenues of 264 feet each.[1]

QUESTIONS:

$\boxed{\textbf{1.}}$ (i.) What is the distance between the red and blue intersection? (ii.) The network of roads and avenues can be generalized to p streets and q avenues and we get the corresponding graph $G_{M-pq} = (V_{M-pq}, E_{M-pq})$ $(|V_{M-pq}| = (p + 1) \cdot$

[1] In nature, the two measures are approximate, but we assume they are correct.

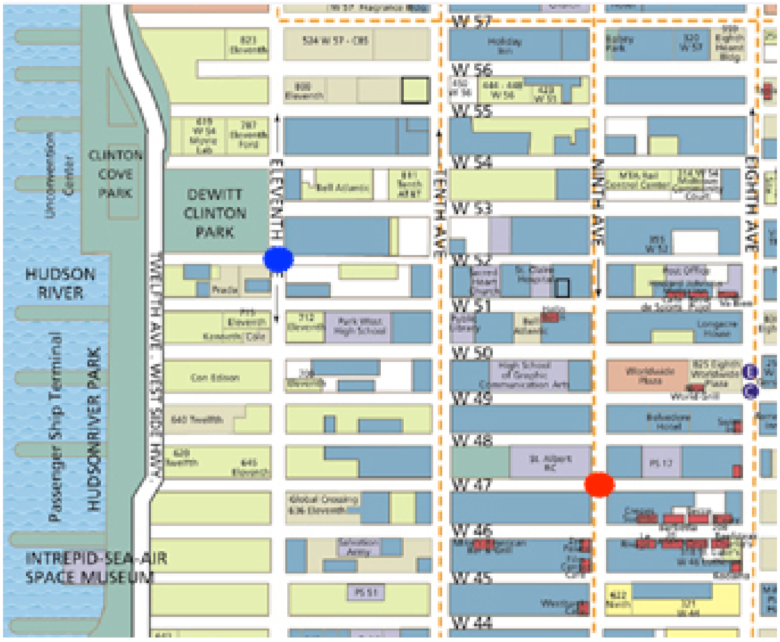

Figure 1: The network of roads and avenues in Manhattan.

$(q+1)$). How large is the set $|E_{M-pq}|$? Justify your answer. (iii.) Write an algorithm that calculates the shortest path from the bottom left intersection (node) to all other intersections (nodes) in the graph G_{M-mn}. (iv.) What is its time complexity? Justify your answer.

2. Despite the great desire of the mayor and extensive road works in Butale, they did not succeed to be completely redesigned along the lines of the New York network. In the end, Peter Puzzle described Butale roads and intersections by the graph $G_B = (V_B, E_B)$, where $|V_B| = n$ and $|E_B| = m$, but the link lengths were 111, 222 or 333 feet. Mathematically, $\forall e \in E_b : w(e) \in \{111, 222, 333\}$. (i.) Write an algorithm that calculates the shortest path from a node $s \in V_B$ to all other nodes in V_B. The time complexity should be $O(n+m)$. (ii.) The well-known Cefizelj drives a taxi in Butale and he figured out how to earn more. Thus, he plans to take passengers from node v to node u not by the shortest but the longest route. Of course, the path must not contain cycles because passengers would notice that. Write an algorithm that in graph G_B for arbitrary nodes u and v finds the longest path without cycles from u to v. (ii.) Show that your solution is correct and what is its time complexity.