Algorithms and data structures 1

Theoretical exam

1. Night appointment

16.01.2023

Name:	
Registration number:	

The information is printed on both sides!

	29 + □	34 + □	22 + □	18 + □	32 + □	19 + □	39 + □
33							

Task 1 [2]

In the table above, enter the digits of your student number in the empty boxes in front of which there is a plus sign.
Do the additions and find the numbers up to . (is already assigned the fixed value 33.)

Task 2 [18]

Write two functions in C++-like notation, each with a parameter n (of type int), whose runtime complexity is of order ÿ(. Use the
master theorem to ^{4} log). One function must be recursive, the other not.
show that the runtime complexity of your recursive function has the desired order.

Task 3 [20]

The values up to . (from task 1) are stored in an array in this order from left to right. Sort the values in ascending order

a) [10] Quicksort (Always use the last - rightmost - value as the pivot element). b) [6] Selection sort c) [4] Merge sort

Specify all the necessary steps in sufficient detail to make it clear how the algorithm works.

Task 4 [20]

a) [9] Insert the values up to from task 1 (in this order) into an initially empty hash table of length 7.

Use hash function ( ) = % and double hashing for collision handling.

The second hash function is ( ) = % + .

Sketch the state of the hash table after each insert step.

(Note: Values can be stored multiple times in the table. The table is static, so it will not be enlarged!)

b) [1] Delete the value from the table and sketch the state of the hash table.

c) [5] Specify the collision path (index positions visited) when searching for the value.

d) [5] Specify the collision path (index positions visited) when searching for the value 50.

Task 5 [20]

a) [8] Insert the values up to from task 1 (in this order) into an initially empty heap. (Values may be stored multiple times in the heap.) Sketch the state of the heap after each insert step.

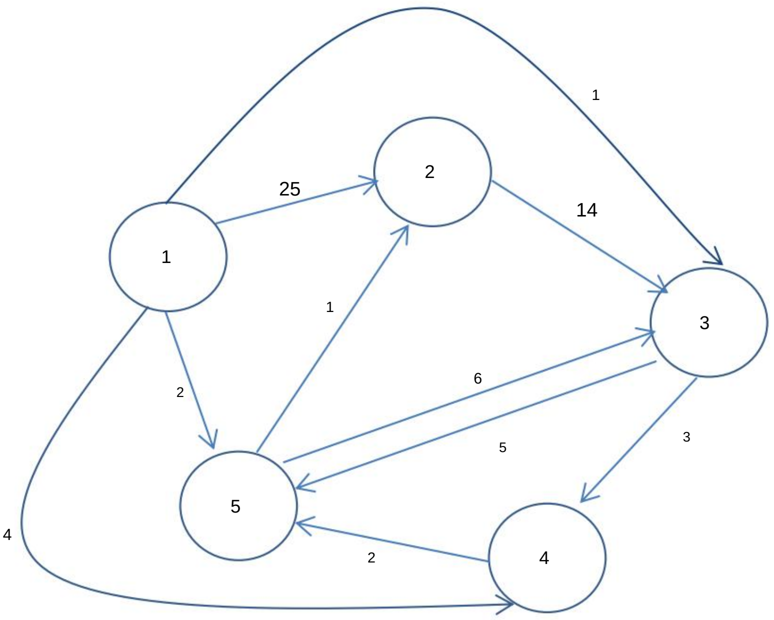
b) [4] In notation similar to C++, give the definition of the most efficient data structure for a heap.

c) [4] In notation similar to C++, give a definition of a function that is as efficient as possible and that determines the depth of the heap in the tree representation is determined and returned.

d) [4] In notation similar to C++, give a definition of an efficient function that determines whether it is a min-heap or a max-heap. The function should return true if it is a min-heap, false otherwise.

Task 6 [20]

The following directed graph is given (the
values to . are to be taken from task 1):



a) [3] Sketch the adjacency list of the graph. b) [10]
Use Dijkstra's algorithm to determine the shortest paths from node 1 to all other nodes
of the graph.

c) [3] Is the graph shown above topologically sortable? If yes, specify a topological sorting, otherwise
explain why one cannot be found.

d) [4] Which of the following requirements is sufficient for the Dijkstra algorithm to deliver the correct result?
Please check the relevant box.

(1) All edge weights of the input graph are non-negative.

(2) The input graph does not contain a negative circle.

(3) The input graph contains a negative circle.

(4) The input graph is a DAG.