Algorithms and data structures 1

Theoretical exam

1. Night appointment

January 17, 2022

Surname:	
Student number:	

The information is printed on both sides!

13 + □	11 + □	32 + □	15 + □	18 + □	23 + □	39 + □	34 + □

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 .

Task 2 [18]

The following functions are given:

```
void g(int i, int n) {
    if (i>0) { for (int
        j=n+10; j>0; j-=5) g(i-2, n);

    }
}

void f(int n) {
    if (!n) return; f(n/(z4%10+2));
    g(z6%5+1, n); for (int i=0;
    i<z7%10; i=i+2)

        f(n/(z4%10+2)); g(z6%5+1,
    n);
}
```

Calculate the runtime complexity of the function f in ÿ notation depending on n. To do this, insert the values (and) determined in exercise 1 for z4, z6 and z7 .

,

(Hint: Create recurrence equations for the travel times of *g* and *f* and solve them using continued substitution or master theorem.)

Task 3 [20]

The values up to . (from exercise 1) **modulo 10** (e.g. %10, %10, etc.) are stored in an array in this order from left to right. Sort the values in ascending order

a) [10] Quicksort
b) [4] Selection sort c)
[6] Heap sort

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

Task 4 [20]

a) [10] 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. b) [2] Delete
the value from the table and sketch the state of the hash table. c) [4] Specify the collision path
(index positions visited) when searching for the value. d) [4] Specify the collision path (index positions
visited) when searching for the value 49.

Task 5 [20]

a) [8] Insert the values up to from task 1 (in this order) into an initially empty binary search tree.
Sketch the state of the tree after each insertion step.
Note: the tree can contain multiple values.

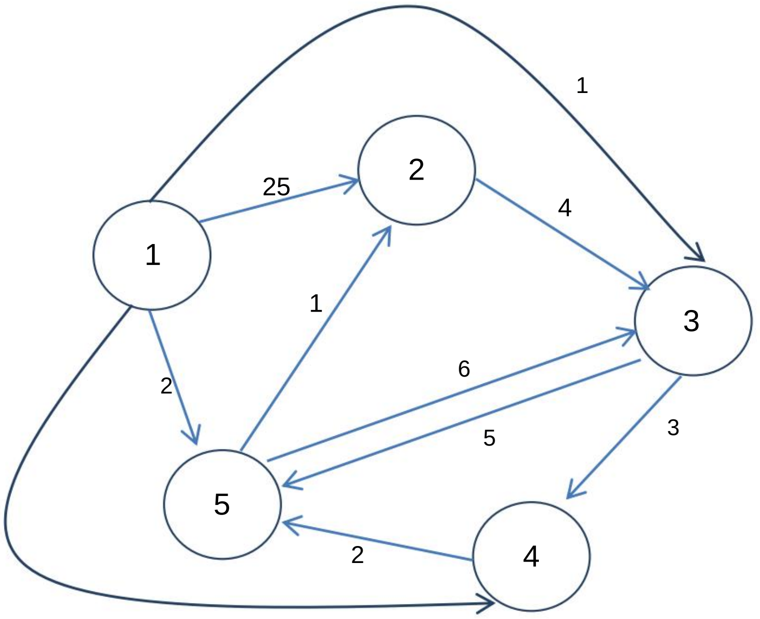
b) [3] In notation similar to C++, give the definition of a data structure for a binary search tree.

c) [5] In notation similar to C++, give a definition of an efficient function or method that checks whether a value (parameter s) is present in the tree. It should return true if successful and false otherwise.

d) [4] Determine the runtime complexity of your function depending on the number (n) of values stored in the search tree in ÿ notation. Briefly justify your result.

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 Graphene.

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.