universität wien

Algorithms and data structures 1

Theoretical exam

Main appointment

12/12/2022

Surname:	
Student number:	

The information is printed on both sides!

+	21 +	38 +	13 +	15 +	40 +	27 +	48 +
45							

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. Carry out the additions and determine the numbers until the fixed value 45 is already assigned). (

Task 2 [18]

Calculate the constants = % + = % + notation, when examining the running time , = % . Give a program in C++ similar with the master theorem the formula parameters a, b and c assume the determined values. Determine the runtime complexity of your program.

Note: If necessary, your program can also consist of several functions that call each other.

universität wien

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] Heap sort c) [4] Bubble sort

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

universität wien

Task 4 [20]

a) [5] 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) [5] 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 a function that returns the maximum and minimum of the in the tree
determines and outputs stored values.
d) [5] Specify which nodes and in which order your function traverses if it is used for subtask a
created tree is called. Is this a preorder traversal, a postorder traversal, an inorder traversal or another type of traversal?

universität wien

Task 5 [20]

a) [10] Add the numbers until (from Example 1) in this order into an originally empty hash table of size 8.
Use the hash function ( ) = % and double hashing with ( ) = as the second hash function for collision handling. Indicate the state of the hash table after each insert operation

b) [4] Specify the collision path that is searched when an attempt is made to insert additionally (from example 1) into the hash table filled after a).

c) [2] Why is the “recoverable” mark used in double hashing?

d) [2] Why is it not a problem in this example that the size of the table (8) is not a prime number?

e) [2] Why is ( ) not an ideal function for collision handling?

universität wien

Task 6 [20]

Given is the following adjacency matrix, which describes the costs of the connections between the nodes of a directed graph:

$$\begin{pmatrix} 0 & 0 & & 4 & \\ & 0 & 0 & 0 & 8 \\ 5 & & & 3 & 2 \\ 2 & 4 & & 1 & 7 \\ 0 & 0 & 3 & 0 & \\ & & & & 5 \\ 0 & 1 & 0 & & 0 \end{pmatrix}$$

a) [2] Sketch the graph described by this adjacency matrix.

b) [10] Use Dijkstra's algorithm to determine the shortest paths from **node 3** to all other nodes of the graph (node **3** corresponds to the node of the third row/column in the adjacency matrix). .) Note all intermediate steps so precisely that it is clear in which order the nodes are visited.

c) [8] Use Kruskal's algorithm to determine a minimally spanning tree of the shadow of the graph. (You get the shadow of the graph by neglecting the directions of the edges. If two nodes are then connected by two or more edges, these edges are combined into one. In other words: Two nodes x and y in the shadow are connected by one if and only then undirected edge connected if at least one of the edges from x to y or from y to x exists in the originally directed graph. They choose the minimum of all as the weight of the undirected edge
directed edges represented by them.) Note all intermediate steps so precisely that it is clear when which edge is added to the spanning tree.

universität wien

universität wien