1st Exam "Algorithms and Data Structures"

February 16, 2023

Name: Chatarina Gopelte

Matriculation Number: 3812071

I hereby declare that I will only use the one-sided handwritten A4 cheat sheet written by me. I feel physically able to take the exam. If I am not registered for the exam, my participation is subject to change. I will not take any copies of the exam out of the room (neither printed nor handwritten).

Signature: Chatarina Gopelte

Score achieved:

1	2	3	~~4~~	5	6	7	8	9	10	Σ
2	3	3	0	2.5	1	1	3	4	1	20.5 ✓

Important Information:

- Please place your ID card and student ID card on the table.
- Turn off and pack away your cell phones completely.
- **You only need to answer 9 of the 10 questions.** If you answer all 10 questions, we will only consider the best 9 of them in the final sum.

  Therefore, you should focus on exactly 9 questions, and only attempt the remaining question if you have time.
- Check that your exam paper includes all 15 pages.
- Write your name on each sheet.
- If you need more paper, we will provide it to you. Do not use your own paper.
- Clearly indicate on new sheets which task you are working on.
- Hand in your cheat sheet along with the exam.
- Do not write with a pencil.
- The total processing time is 120 minutes.

Good luck!

NAME:

Aufgabe 1: Landau Notation and Growth of Functions (2.5 + 0.5 + 1 Punkte)

a) Decide for each of the following function pairs f and g, in which asymptotic relationship they stand to each other. Cross out EVERY box that represents a true statement. You receive 0.5 points for every completely correctly filled row in the table.

f	g	$f \in o(g)$	$f \in \mathcal{O}(g)$	$f \in \Theta(g)$	$f \in \Omega(g)$	$f \in \omega(g)$
$sin(n)$	n	☒	☒	☐	■ ☐	☐
$n^2 + cos(n)$	$n^2 + \frac{2}{n}$	☒	☒	☐	☒	☐
$(\sqrt{5})^n$	4^n	☒	☐	☐	☒	☐
$\log(n^n)$	$n^{2/3}$	☐	☐	☒	☒	☐
$n \mod 20$	n^2	☒	☒	☐	☐	☐

no cross!

b) Which of the following statements implies that $f(n) \in \Omega(g(n))$ holds? Cross exactly one box.

☒ $f(n) \leq 4g(n)$ for all $n \geq 1$

☐ $f(n) \geq 4g(n)$ for all $n \geq 136$

☐ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

☐ None of the above statements.

c) Prove that $\sqrt{n} \in o(n^{2/3})$ holds.

To prove that $\sqrt{n} \in o(n^{2/3})$, we need to show that

$$\lim_{n\to\infty} \frac{\sqrt{n}}{n^{2/3}} = 0$$

We can simplify the expression as follows

$$\lim_{n\to\infty} \frac{\sqrt{n}}{n^{2/3}} = \lim_{n\to\infty} \frac{n^{1/2}}{n^{2/3}} = \lim_{n\to\infty} n^{-1/6} = 0$$

Since the limit evaluates to 0, we have proven that $\sqrt{n} \in o(n^{2/3})$.

NAME:

Aufgabe 2: Multiple Choice - Diverse Topics (Score per sub-task: 0.5 Punkte)

For all questions, exactly one answer choice is correct and therefore only one box is to be filled in. If you fill in more than one box for a question, your answer for that question will be considered incorrect. For each correctly answered question, you will receive 0.5 points, for an incorrectly answered question 0 points.

a) If we could find the pivot element in the QuickSort algorithm in $O(1)$ time, then we would get the best guarantee on the running time with

☐ the arithmetic mean ☒ the median

b) The connected components of an undirected graph are uniquely determined

☒ Correct ☐ Incorrect

c) The principle of "Memoization" in dynamic programming means

☐ we store the dynamic-programming recurrence in a static array so that it may be applied at each level of recursion.

☒ we save the solutions of smaller subproblems if we need them again.

☐ the dynamic-programming algorithm can remember the current location in the recursion tree.

☐ we will remember the recursions occurring in dynamic programming for future exams.

d) Which of the following properties is *not* a desired property of a hash function $h(x)$?

☐ $h(x)$ should be computable in $O(1)$ time.

☐ The values of $h(x)$ should be within the size of the hash table.

☐ For different x, $h(x)$ should also assume different values.

☒ If x_1 to x_n are the elements to be inserted into the hash table, then $h(x_1)$ to $h(x_n)$ should be evenly distributed over the natural numbers.

e) In the Relax-operation of the Floyd-Warshall algorithm, if we replace the minimum with the maximum, then the algorithm will not find the shortest, but the longest paths between pairs of vertices.

☒ Correct ☐ Incorrect

f) If you increase every edge weight by the same positive constant, then the solution to the single-source shortest-path problem will not change.

☒ Correct ☐ Incorrect

g) A connected, unweighted Graph $G = (V, E)$ has a unique minimum spanning tree.

☐ Correct ☒ Incorrect

h) If $T(n) = 2T(n/3) + \sqrt{n}$ holds, then we also have

☐ $T(n) = \Theta(\sqrt{n})$

☐ $T(n) = \Theta(n^2)$

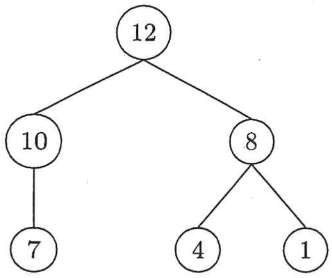
☒ $T(n) = \Theta(n^{\log_3(2)})$

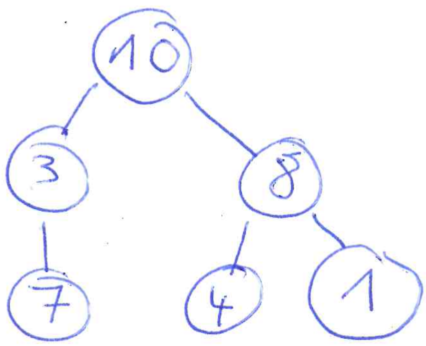
☐ $T(n) = \Theta(n^{\log(3)})$

NAME:

Aufgabe 3: Data structures (1 + 1 + 0.5 + 1.5 Punkte)

a) Perform the DecreaseKey(0, 3) operation on the following Max-Heap (in this operation, the key of the root element is set to 3). Draw the resulting Max-Heap.





✓ 1

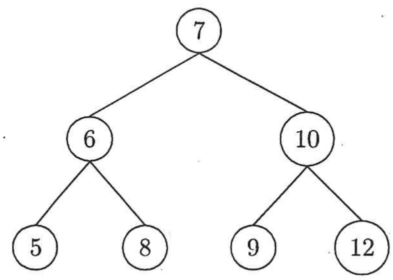
b) Give an instance of a knapsack problem with 4 different items and weights where the greedy algorithm does not find the optimal solution. The greedy algorithm selects in each step the item with the best value-weight ratio that still fits in the knapsack.

Item	Weight	Value
1	3	8
2	2	7
3	4	9
4	5	10

Knapsack capacity: 7

Lösungsweg? nicht ersichtlich 0,~~5~~

c) Is the following tree a binary search tree?



☐ Yes

☒ No ✓

0,5

d) Let T be a binary search tree with root node x. It is assumed that access to the children of any node can be obtained with $x.left$ and $x.right$. It holds that $x.left == NIL$ or $x.right == NIL$ if no left or right subtree exists. Provide the pseudo-code for an algorithm that outputs the nodes of the binary search tree in sorted order from largest to smallest using $PRINT$.

function PRINTBSTREVERSESORTED(Root x)

```
if x != NIL then
    PRINTBSTREVERSESORTED(x.right)
    PRINT(x.key)
    PRINTBSTREVERSESORTED(x.left)
ENDIF
```

✓

end function

1,5

3

NAME:

Aufgabe 4: Hashing and MST (1 + 3 Punkte)

a) Consider hashing with linear probing and the hash function $h(x) = (7 \cdot x + 2) \mod 11$. The corresponding hash table consists of an array of length 11. Insert the elements 2, 3, 7, 15 and 1 in this order into the hash table.

-1p

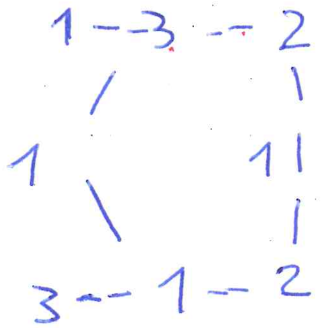
15	7	2	3	1						

b) Consider the following algorithm. The algorithm receives as input a connected, undirected, weighted graph $G = (V, E, c)$ with edge weight function $c : E \to \mathbb{R}_{\geq 0}$.

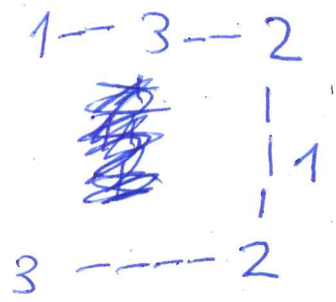
function ALTERNATIVEMST
    $E_T = \emptyset$
    Sei s beliebiger Knoten in G
    for $v \in V \setminus \{s\}$ **do**
        Let $e = (v, w)$ be the edge of v with minimal weight $c(e)$ that does not belong to E_T
        $E_T \leftarrow E_T \cup \{e\}$
    end for
    return (V, E_T)
end function

Is AlternativeMST correct, that is, does AlternativeMST actually calculate a Minimum Spanning Tree of G? Prove or disprove that AlternativeMST calculates a MST.

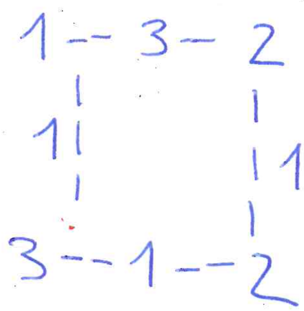
Alternative MST is not correct and does not always calculate an MST. Here is a counterexample

G = 

Running AlternativeMST on G ~~would~~ with s = 1 would produce



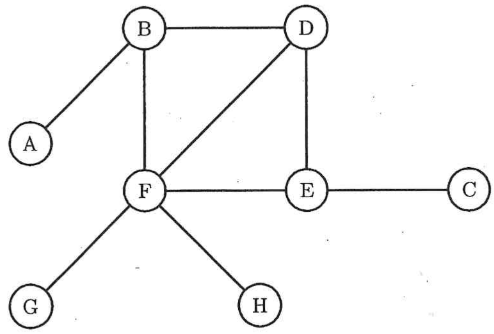
However, the actual MST would be



2

NAME:

Aufgabe 5: DFS (1 + 1.5 + 0.5 + 0.5 + 0.5 Punkte)



a) Perform the DFS algorithm on the above graph, starting at node A. Assume that the DFS algorithm visits nodes with smaller letters first when there is a choice. Give the order in which the various nodes are visited.

A B D F E C G H f

b) Now give the discovery time and the finishing time of the different nodes. Start with the discovery time $t = 1$ for node A.

Ff.

Knoten v	Discovery-Zeit(v)	Finishing-Zeit(v)
A	1	16
B	2	9
C	14	15
D	3	8
E	10	13
F	4	7
G	11	12
H	5	6

$\frac{1}{1{,}5}$

c) An undirected and unweighted graph is given as an adjacency matrix. Then the running time of the DFS algorithm is $O(|V| + |E|)$.

☐ Correct ☒ Incorrect ✓

d) Let $G = (V, E)$ be a directed graph. Which of the following algorithmic problems can be solved with DFS? Put exactly one cross.

☐ Finding a source node.

☐ Finding a sink node. ✓

☒ Both problems.

☐ Neither of the two problems.

e) Let $G = (V, E)$ be a directed, unweighted graph. A topological sorting exists only if the graph is acyclic.

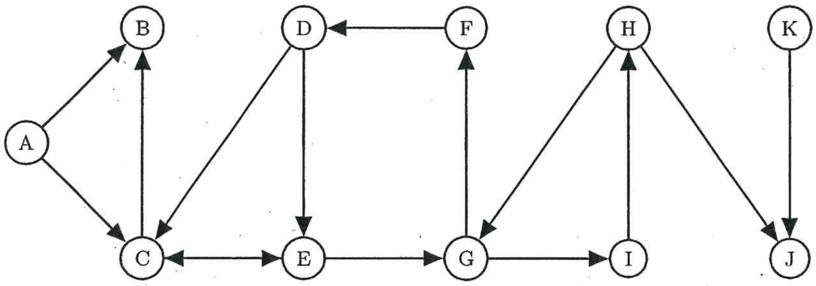
☒ Correct ☐ Incorrect ✓

$\frac{2{,}5}{4}$

NAME:

Aufgabe 6: Connected Components (0.5 + 1 + 1 + 1 + 0.5 Punkte)

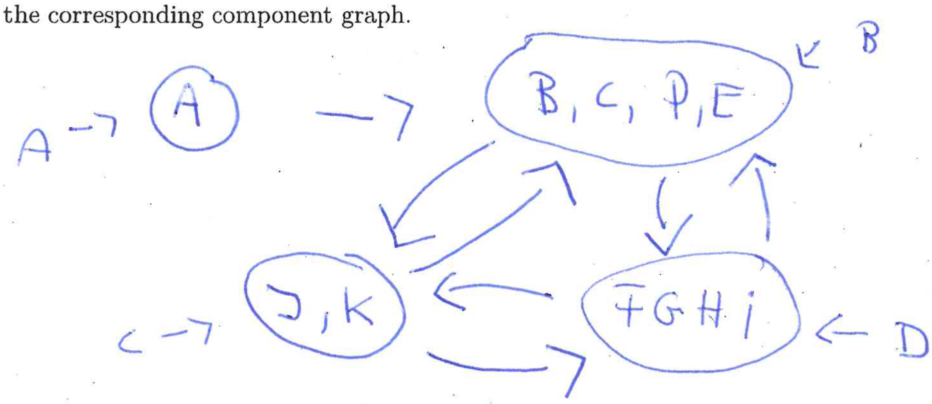
Consider the following directed Graph.



a) How many strongly connected components does this graph have?

4

b) Draw the corresponding component graph.



c) Number the different nodes of the above component graph according to the alphabet, i.e., A, B, C, etc. Then specify a topological sort of the component graph.

A C D B

d) Which nodes of the above component graph are source nodes, which are sink nodes?

Folgefehler

Source nodes: A

0,5

Sink nodes: C

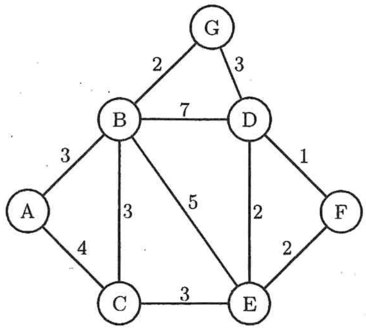
e) In a directed, acyclic graph, the component graph corresponds to the original graph.

☒ Correct ☐ Incorrect

1/4 Pkt.

NAME:

Aufgabe 7: Shortest Paths (1+1+0.5+0.5+1 Punkte)



a) Perform the Dijkstra algorithm on the above graph to find the shortest path from A to all other nodes. Give the order in which the Dijkstra algorithm visits the nodes.

A B G C E D F

b) Now give the length of the shortest path from A to all other nodes.

B	C	D	E	F	G
3	~~6~~ 4	~~10~~ 8	~~9~~ 7	~~11~~ 9	5

c) If a graph does not contain a negative weight cycle, then the result of the Dijkstra algorithm is correct.

☒ Correct ☐ Incorrect

d) For graphs with $\Theta(|V|^2)$ many edges, the Dijkstra algorithm is best implemented using a

☐ Array. ☒ Heap.

e) Let $G = (V, E)$ be a directed graph. Now assume we define a weight function on the nodes, i.e., $c : V \to \mathbb{R}_{\geq 0}$. Does it still make sense to speak of the shortest path problem with a weight function on the nodes? If Yes, then give a formal definition of the problem. If No, then explain why the problem makes no sense.

It is not meaningful to speak of the shortest path problem with a weight function on the nodes, as the weight function is typically defined on the edges in the problem. The shortes paths problem seeks to find the shortest path between two nodes in a graph, where the distance between nodes is defined by the sum of the weights of the edges on the path between them.

1 BE

see other side

However, if we define a weight function on the nodes, we can modify the shotest path problem to account for this by incorporating the node weights into the edge weights. Specifically, we can define the weight of an edge (u,v) to be the sum of the weight of the starting node u and the weight of the ending node v, i.e. $w(u,v) = c(u) + c(v)$. Then, we can solve the modified shortest path problem using any standard shortest path algorithm, such as Dijkstra's algorithm or Bellman-Ford algorithm.

seen

NAME:

Aufgabe 8: Algorithms (0.5+0.5+1+0.5+0.5+1 Punkte)

a) What is the Best-Case running time of the following algorithm in relation to the length n of the input array A?

```
function ArrayFunction(Array A)
    for i=1,...,A.length do
        A[i] = i
    end for
    A = QuickSortWithLastElementAsPivot(A)
    return A
end function
```

Best-Case: $\Theta($ n log n $)$

f

b) The following program calculates a hash function. Which hash function is calculated by the program?

```
function HashFunction(Integer n)
    n = n + 2
    n = 7 * n
    while n ≥ 153 do
        n = n − 153
    end while
    return n
end function
```

Hashfunktion: $\boxed{7 \cdot (n+2) \bmod 153}$

✓

c) What are the Best-Case and Worst-Case running time of the following algorithm in relation to the length n of the input array A?

```
function ArrayFunction(Array A)
    for j = 2 to A.length do
        key = A[j]
        i = j − 1
        while i > 0 and A[i]>key do
            A[i + 1] = A[i]
            i = i − 1
        end while
        A[i + 1] = key
    end for
    return A
end function
```

Best-Case: $\Theta($ n $)$

Worst-Case: $\Theta($ n^2 $)$

✓

d) What is a Worst-Case input sequence of length n for the algorithm in c)?

$n, n-1, \dots, 3, 2, 1$

✓

(Continuation of Exercise)

e) What is the Worst-Case running time of the following algorithm in relation to the length n of the input array A?

```
function MyAlgorithm(Array A)
    if A.length ≤ 100 then
        return A[1]
    end if
    Result = 4 · MyAlgorithm(A[1 ... A.length-1])
    return Result
end function
```

Worst-Case: $\Theta($ n $)$

✓

Hint: A[1 ... A.length-1] is the Array A without the last element. This operation takes time $O(1)$.

f) The diameter of an undirected and unweighted graph corresponds to the largest distance between two nodes in a graph. More specifically, the diameter is

$$\mathrm{diam}(G) = \max_{(u,v)\in V\times V} \{\mathrm{dist}(u,v)\},$$

where $\mathrm{dist}(u,v)$ corresponds to the length of the shortest path between u and v. Provide an algorithm that calculates the diameter of a graph $G = (V, E)$. Describe your algorithm in pseudo-code.

```
Function diameter (graph G):
    max_distance = 0
    for each vertex v in G:
        distances = BFS (G, v)    // or DFS(G, v)
        max_distance = max(max_distance,
                           max(distances))
    return max_distance
```

-0,5

This Algorithm iterates through all vertices in the Graph and performs BFS to find the shortest path to all other vertices. It then updates the maximum distance found so far, and returns it as the diameter of the graph.

NAME:

Aufgabe 9: Matrix Travel (2+2 Punkte)

Given an MxN matrix A of positive numbers, we can imagine this matrix as a game board of size MxN. The entry at (i, j) corresponds to the costs that must be paid to enter field (i, j) of the game board. We now want to move a game piece as cost-effectively as possible from the upper left field $(1, 1)$ to the lower right field (N, M). In each step, the game piece may only move one field down or one field to the right. So it can go from field (i, j) to field $(i + 1, j)$ or to field $(i, j + 1)$. We are interested in the costs of the cheapest path from $(1, 1)$ to (N, M). In the following example, the cheapest path costs 33.

$$A = \begin{pmatrix} 4 & 7 & 8 & 4 & 6 \\ 6 & 7 & 3 & 9 & 2 \\ 3 & 8 & 1 & 2 & 4 \\ 7 & 1 & 7 & 3 & 7 \\ 2 & 9 & 8 & 9 & 3 \end{pmatrix}, \qquad \text{A cheapest path } A = \begin{pmatrix} 1 & 7 & 8 & 4 & 6 \\ 6 & 7 & 3 & 9 & 2 \\ 3 & 8 & 1 & 2 & 4 \\ 7 & 1 & 7 & 3 & 7 \\ 2 & 9 & 8 & 9 & 3 \end{pmatrix}$$

a) This problem can be solved with dynamic programming. Set up an appropriate recursive equation.

T is minimum cost to reach the cell $\rightarrow$ $T[1,1] = A[1,1]$, $T[i,1] = \sum_{k=1}^{i} A[k,1]$, $T[1,j] = \sum_{k=1}^{j} A[1,k]$

For $i,j > 1$:

$T[i,j] = A[i,j] + \min(T[i-1,j], T[i,j-1])$ ✓

b) Complete the following pseudo-code to solve the problem in runtime $O(MN)$.

```
function MatrixTravel(Cost matrix A)
    T = MxN matrix filled with zeros
    for i = 1 to N do
        for j = 1 to M do
            if i=1 and j=1:
                T(1,1) = A(1,1)
            elsif i=1:
                T(1,j) = A(1,j) + T(1,j-1)
            elsif j=1:
                T(i,1) = A(i,1) + T(i-1,1)
            else:
                T(i,j) = A(i,j) + min(T(i-1,j), T(i,j-1))
            endif
        end for
    end for
    return T[N,M]
end function
```

✓

NAME:

Aufgabe 10: Prove or Disprove (4 Punkte)

a) Let $G = (V, E)$ be a connected, undirected and weighted graph. A cut $(S, V \setminus S)$ is a partition of the nodes (that is, we divide the nodes into two groups, not necessarily of equal size). A cut edge is an edge (u, v) with one node in S and one node in $V \setminus S$, that is $u \in S$ and $v \in V \setminus S$. Prove or disprove the following statement:

Assume that G has a uniquely determined minimum spanning tree. Then, for each cut $(S, V \setminus S)$ of G, there exists a uniquely determined cut edge with minimum weight.

The statement is true.

Proof: Let T be the unique ~~MST~~ MST of G, consider any cut $(S, V\setminus S)$ of G. We want to prove that there exists a unique edge with minimum weight that crosses this cut.

Assume for contradiction that there are two different cut edges (u,v) and (x,y) with the same minimum weight. W.l.o.g., assume $u, x \in S$.

There exists a path in T between u and x. Let z be the first node on this path that is in $V\setminus S$. Then, the edge (z,x) must be in the unique path in T between u and x. Therefore, the weight of of (z,x) is less or equal ~~to~~ the weight of (u,v) wich is the same as the weight of (x,y). This contradicts the assumption that (u,v) and (x,y) have the same minimum weight.

Therefore, there can be at most one cut edge with minimum weight that crosses the cut $(S, V\setminus S)$. To show that there exists such an edge, we can consider the unique path in ~~MST~~ T between any ~~~~ node in S and any node in $V\setminus S$. This path must contain at least one edge that crosses the cut $(S, V\setminus S)$.

↪ next page

NAME:

(additional Page)

and the edge with minimum weight among those edges is the unique cut edge with minimum weight that crosses the cut.

Therefore, for each cut (S, V\S) of G, there exist a uniquely determined cut edge, as required.

1/4 Pkt.