University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn

Algorithms and Data Structures Exam

12th march 2025, 14:00 -17:00

Name: ....

Matriculation No.: ....

Signature: ....

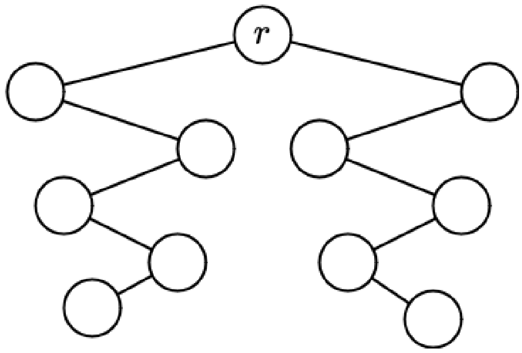
Do not open or turn until told so by the supervisor!

- Put your **student ID** in front of you or on the table next to you.
- Write your **name** and **matriculation number** on this page and **sign** the document.
- Your **signature** confirms that you have answered all exam questions yourself without any help, and that you have notified exam supervision of any interference.
- You are allowed to use a summary of **six handwritten, single-sided A4 pages**.
- **No electronic devices** are allowed.
- Write legibly and only use a pen (ink or ball point). Do **not use red**! Do **not use a pencil**!
- You may write your answers in **English or German** language.
- Only **one solution per task** is considered! Make sure to strike out alternative solutions, otherwise the one yielding the minimal number of points is considered.
- **Detailed steps** might help you to get more points in case your final result is incorrect.
- The keywords **Show...**, **Prove...**, **Explain...** or **Argue...** indicate that you need to prove or explain your answer carefully and in sufficient detail.
- The keywords **Give...**, **State...** or **Describe...** indicate that you need to provide an answer solving the task at hand but without proof or deep explanation (except when stated otherwise).
- You may use information given in a **Hint** without further explanation.
- **Read each task thoroughly** and make sure you understand what is expected from you.
- **Raise your hand** if you have a question regarding the formulation of a task or if you need additional sheets of paper.
- A total of **45 points** is sufficient to pass and a total of **90 points** is sufficient for the best grade.
- Write your name on **all sheets**!

Task	1	2	3	4	5	6	7	**Total**
Maximum	25	15	10	20	15	20	15	120
Points								

Task 1: Short Questions *(25 Points)*

(a) Give a sequence of insert operations for an (unbalanced) binary tree datastructure, that results in the tree structure depicted below. (So it should be exactly 11 insert operations.) *(4 Points)*



(b) Prove or disprove: A binary search tree can have depth $o(\log n)$, where n is the number of elements in the binary search tree. *(5 Points)*

(c) Give an algorithm that decides for a given connected, directed, weighted graph $G = (V, E, w)$ whether G contains a positive cycle. All edge weights are real numbers ($w : E \to \mathbb{R}$). We call a cycle positive, if the sum of the weights of its edges is strictly positive. Argue why your algorithm is correct. What is the asymptotic complexity of your algorithm? *(4 Points)*

(d) What is the asymptotic complexity of the following pseudocode? Give the complexity in terms of $\Theta(\cdot)$ notation and give an explanation.

Algorithm 1 `Mystery`(n)

```
for i = n/10 to n do
    y ← i
    while y < n do
        y ← y · 2 + 1
```

(5 Points)

(e) Given a weighted graph $G = (V, E, w)$, with $w : E \to \mathbb{R}$, and $T \subset E$, a Minimum Spanning Tree of G. Prove that T is also an MST of $G' = (V, E, w')$, where $w'(e) = (w(e))^3$. *(7 Points)*

Solution Task 1

Task 2: Landau-Notation *(15 Points)*

(a) Below are 5 functions in $n \in \mathbb{N}$. Give a sorted order of these functions in terms of asymptotic growth. That is if f appears before g in the sorted order, then $f(n) \in \mathcal{O}(g(n))$. You are only required to give the correct order, proofs are not necessary. *(5 Points)*

- $a(n) = 20^{20^{20^{20^{23456789876543456789098765434567890987654356789098765434567890987654345678909876545 60}}}}$
- $b(n) = 2^n \cdot \log n$
- $c(n) = 16^{\log_2(n)} + 3n^2$
- $d(n) = 78n^3$
- $e(n) = 4^{n/2}$

(b) Prove or disprove based on the definition of the Landau-Notation:

$$n^{1/4} \cdot n^{1/3} \cdot n^{1/2} \in \Theta(n)$$

(4 Points)

(c) Prove using the definition of the Landau-Notation:

$$\frac{64}{6} \cdot \left\lceil \frac{n}{2} \right\rceil ! \in \Omega(2^n)$$

You may assume that n is always even. *(6 Points)*

Solution Task 2

Task 3: String Isomorphisms *(10 Points)*

Let $S_1 = c_1c_2c_3c_4\ldots c_n$ and $S_2 = d_1d_2\ldots d_n$ be two strings of the same length. C is the set of all characters that appear in S_1, and D is the set of all characters that appear in S_2. We say that S_1 and S_2 are **isomorphic** if there exists a bijective function $f : C \to D$ such that for all $1 \leq i \leq n$, we have $f(c_i) = d_i$.

Example: Let S_1 = "paper" and S_2 = "title". Then the sets of unique characters are: $C = \{a, e, p, r\}$ and $D = \{e, i, l, t\}$.

A valid function f can be defined as: $f(\text{p}) = \text{t}, f(\text{a}) = \text{i}, f(\text{e}) = \text{l}, f(\text{r}) = \text{e}$, which proves that S_1 and S_2 are isomorphic.

Design an efficient algorithm that, given two strings S_1 and S_2, determines whether they are isomorphic. If they are, your algorithm must output a valid isomorphism.

Argue why your algorithm is correct and analyze its runtime in terms of the length n of the two strings.

Solution Task 3

Task 4: A Thiefs Plan *(20 Points)*

There are n houses built in a line, each of which contains some money $m_1, m_2, \dots, m_n$ in it. A robber wants to steal money from these houses, but they can't steal from two adjacent houses (e.g. they could steal from 2 and 4, but not from 3 and 4).
Give an efficient algorithm that takes as input the values $m_1, m_2 \dots, m_n$ and outputs a set of house-numbers $i_1, i_2, \dots, i_k$, such that the collected money

$$\sum_{1 \leq j \leq k} m_{i_j}$$

is maximised. Furthermore, no two consecutive numbers are contained among $i_1, i_2, \dots, i_k$.
What is the asymptotic complexity of your algorithm? Give an explanation.

Solution Task 4

Task 5: Hashing with Open Addressing *(15 Points)*

We consider hash tables with open addressing and two methods for resolving collisions: double hashing and cuckoo hashing. Let m be the size of the hash table. We define

$$h_1(x) := (5 \cdot x) \mod m,$$
$$h_2(x) := 1 + (2x \mod (m-1)),$$
$$h_3(x) := (3 \cdot x - 2) \mod m.$$

(a) Let $h_d(x,i) := (h_1(x) + i \cdot h_2(x)) \mod m$. Insert the keys 13, 14, 2, 3, 11 sequentially into a hash table of size $m := 11$. Use h_d and double hashing for collision resolution. *(5 Points)*

(b) Insert the values 3, 10, 7 sequentially into a hash table of size $m := 7$. Use cuckoo hashing with the functions h_1 and h_3 for collision resolution. Provide the intermediate state of the table after each insertion (i.e., three tables in total). *(5 Points)*

(c) Prove or disprove: For any $m \in \mathbb{N}$ and universe $\mathbb{N}$, it holds that the set of hashfunctions $H = \{h_i(x) = x \cdot i \mod m \mid 0 \leq i \leq m-1\}$ is 2-universal. *(5 Points)*

Note: Write your solutions for (a) and (b) in the tables on the solution sheet provided for this question.

Solution Task 5

Task (a):
Hashtable after inserting all elements:

0	1	2	3	4	5	6	7	8	9	10

Task (b):
After inserting 3:

0	1	2	3	4	5	6

After inserting 10:

0	1	2	3	4	5	6

After inserting 7:

0	1	2	3	4	5	6

Task 6: k-near *(20 Points)*

Given a (undirected) graph $G = (V, E)$ and two nodes $u, v \in V$, we are interested to determine wether or not u and v are at most $k \in \mathbb{N}$ hops apart.

(a) Give an algorithm to determine whether two given nodes u, v are at most k hops apart.

The input to your algorithm consists of the graph $G = (V, E)$, a parameter $k \in \mathbb{N}$ and two nodes $u, v \in V$.

The algorithm must run in linear time (i.e. the algorithm must terminate after at most $O(n + m)$ operations).

Argue why your algorithm is correct and analyze the running time. *(5 Points)*

We now change the graph a bit, we now give weights to the edges. Each edge either has weight 1, or weight 0. So we are now given a weighted graph $G = (V, E, w)$ with $w : E \to \{0, 1\}$. We are now interested in determining whether, or not two nodes u, v are at distance at most k.

(b) Give an algorithm to determine whether two given nodes u, v have distance at most k.

The input to your algorithm consists of the graph $G = (V, E, w)$, a parameter $k \in \mathbb{N}$ and two nodes $u, v \in V$.

The algorithm must run in linear time (i.e. the algorithm must terminate after at most $O(n + m)$ operations).

Argue why your algorithm is correct and analyze the running time. *(15 Points)*

Solution Task 6

Task 7: Rotated Text *(15 Points)*

Given two string $C = c_0c_1 \ldots c_{n-1}$ and $D = d_0d_1 \ldots d_{n-1}$ both of length n, we say that D is a rotation of C if there exists a value j, such that for all $0 \leq i < n$ it holds that $c_i = d_{i+j \mod n}$. In other words D is C, but circularly shifted by j.

Example: The string "thmAlgori" is a rotation of "Algorithm", and in this case $j = 3$. On the other hand "lgoAritmh" is not a rotation of "Algorithm" (it is a permutation, but it is not a rotation).

Give an algorithm that takes as input two strings of length n and determines in $O(n)$ time, whether one is a rotation of the other. Prove that your algorithm is correct and that it satisfies the asymptotic running time constraints.

Solution Task 7