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

Algorithms and Data Structures Exam

24. August 2022, 14:00 -17:00

Name: ....

Matriculation No.: ....

Signature: ....

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

- 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 **five 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	8	**Total**
Maximum	22	12	12	15	15	12	12	20	120
Points									

Task 1: Short Questions *(22 Points)*

(a) The goal is to store the keys $k_1 = 12$, $k_2 = 15$, and $k_3 = 8$ in a hash table of size $m = 7$ using the variant of *Cuckoo Hashing* described in the lecture. The two hash functions are defined as:

$$h_1(x) = 2 \cdot x + 1 \mod m, \quad h_2(x) = x + 3 \mod m.$$

Fill in the left table with the state of the hash table after inserting k_1 and k_2. In the right table, show the final state after inserting k_3. *(4 Points)*

0	1	2	3	4	5	6

0	1	2	3	4	5	6

(b) Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous function with $f(0) = -1$ and $f(1) = 1$. Describe an algorithm that finds a root of f with a precision of $1/n$ in $O(\log n)$ time. The algorithm should return a number x such that f has a root in the interval $(x - 1/n, x + 1/n)$. Assume that evaluating $f(x)$ takes constant time. Justify the correctness of your algorithm. *(6 Points)*

(c) Assume there is a priority queue H with the following operation runtimes:

- `create`, `insert`, `getMin`, and `decreaseKey`: $O(1)$
- `deleteMin`: $O(\log n)$ (where n is the number of elements in the data structure)

What is the asymptotic runtime of Dijkstra's Algorithm (as a function of the number of edges m and the number of vertices n) when using H as the underlying priority queue? Justify your answer. *(5 Points)*

(d) Prove or disprove the following statement: Heapsort is stable. *(7 Points)*

Hint: Assume Heapsort uses the array-based binary heap implementation from the lecture. A sorting algorithm is considered stable if the initial relative order of equal elements is preserved. For example, if the initial array consists of the key-value pairs $[(3, a), (1, r), (1, b)]$, then $[(1, r), (1, b), (3, a)]$ is a stable sorting, but $[(1, b), (1, r), (3, a)]$ is not.

Solution Task 1

Task 2: Landau-Notation *(12 Points)*

(a) Consider the following five functions in terms of $n \in \mathbb{N}$. Provide a sorted order of these functions with respect to Big-O notation, such that for two consecutive functions f and g in this order, $f(n) \in O(g(n))$. You do not need to justify your answer. *(4 Points)*

- $a(n) = 3 \cdot n^2 \cdot \log_2(n) + 4$
- $b(n) = 2^{2 \log_2(n)}$
- $c(n) = \frac{a(n)}{\sqrt{n+1}}$
- $d(n) = \log_2(n!)$
- $e(n) = \sqrt{n^5}$

(b) Prove or disprove the following statement using the definition of Big-O notation:

$$10 \cdot n^9 \in \Omega(n^{10})$$

(4 Points)

(c) Prove or disprove the following statement using either the definition of Big-O notation or the limit-based characterization:

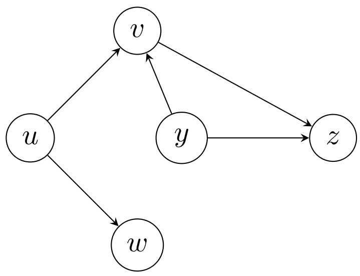
$$\sum_{i=0}^{n} (2 \cdot (i+1)) \in o(n^3)$$

(4 Points)

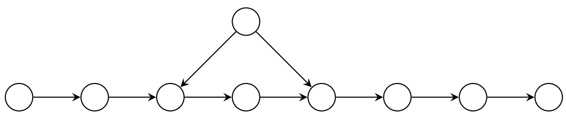
Solution Task 2

Task 3: Topological Sort *(12 Points)*

(a) Provide a topological sorting of the following graph. *(3 Points)*



(b) How many topological orderings are there for the given graph? *(3 Points)*



(c) Let $G = (V, E)$ and $G' = (V, E')$ be two Directed Acyclic Graphs (DAGs) with the same vertex set V, where G is a subgraph of G' (i.e., $E \subseteq E'$). In which graph are there more possible topological orderings? Justify your answer. *(6 Points)*

Solution Task 3

Task 4: Rabin-Karp with DigitSums *(15 Points)*

In this task, we work with a modified version of the Rabin-Karp algorithm. Let $x = x_1x_2x_3 \dots x_r$ represent the decimal representation (base 10) of x. The digit sum of x is defined as:

$$\sum_{i=1}^{r} x_i$$

For example: $1927 = 1 + 9 + 2 + 7 = 19$. The hash function used is:

$$h(x) = x \mod M$$

If the hash value of the pattern matches the hash value of a substring, a TestPosition function is used to verify the actual match.

(a) Let the search text be $T = 12518230051251$ (length $n = 14$) and the pattern be $P = 251$ (length $m = 3$). The hash function used is:

$$h(x) = x \mod 5$$

Provide the hash value $h(T[s, s+1, s+2])$ for all $s \in \{0, 1, \dots, 11\}$. *(9 Points)*

(b) How many times does the TestPosition function need to be called for the above example? *(2 Points)*

(c) Provide a method to compute $h(T[s+1, s+2, \dots, s+m])$ in constant time, given that the hash value of the previous substring $h(T[s, s+1, \dots, s+m-1])$ is already known. *(4 Points)*

Hint: Assume that addition, subtraction, multiplication, division, and modulo operations all take constant time.

Solution Task 4

Hashvalues:

s	$T[s, s+1, s+2]$	$h(T[s, s+1, s+2])$
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

Task 5: Longest Common Substring *(15 Points)*

Given two strings A and B of length n over the alphabet $\{0, 1\}$. We want to compute the length of the longest common substring of A and B. A substring is defined as a contiguous sequence of characters in A or B.

(a) Provide an algorithm that solves this problem in $O(n^2)$ time and justify its runtime. *(15 Points)*

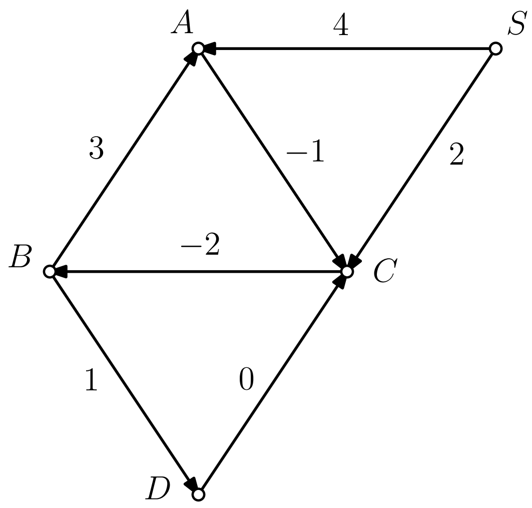
Alternative: You may provide an $O(n^3)$ algorithm instead, but you will receive a maximum of 5 points.

Hint: There exists a solution using bottom-up dynamic programming. Keep track of the maximum length of a common substring ending at indices i and j in A and B, respectively.

Solution Task 5

Task 6: Shortest Paths *(12 Points)*

Consider the following directed graph with $n = 5$ nodes as shown in the diagram.



(a) Does this graph contain a negative cycle? If yes, mark the edges of one such cycle (circle or highlight the edges). *(2 Points)*

(b) Run the Bellman-Ford Algorithm on this graph starting from node S. After each iteration of the outer loop, fill in the currently computed distances from S in the provided table. *(10 Points)*

Important: In the inner loop of the Bellman-Ford Algorithm, iterate over the edges in ascending order of their weights, i.e., in the order $-2, -1, 0, 1, 2, 3, 4$.

Solution Task 6

Solutiontable:

	$\delta(S,S)$	$\delta(S,A)$	$\delta(S,B)$	$\delta(S,C)$	$\delta(S,D)$
Initial	0	∞	∞	∞	∞
$i=1$					
$i=2$					
$i=3$					
$i=4$					

Task 7: MysteryAlgorithm *(12 Points)*

Let $G = (V, E)$ be an undirected graph with n nodes. Consider the following algorithm Mystery given in pseudocode, which takes as input G and a node $s \in V$.

Algorithm 1 `Mystery`(G, s)

```
D ← a new Dictionary
D[s] ← 0
for every node u ∈ V \ {s} do
    D[u] ← −1
u ← s; i ← 1
while there exists an edge {u, v} ∈ E, such that D[v] = −1 do
    v ← an arbitrary neighbor u with D[v] = −1
    D[v] ← D[u] + i
    u ← v; i ← i + 1
```

(a) What is the maximum number of iterations the while-loop can run as a function of n? *(6 Points)*

(b) What is the maximum possible value in D after executing Mystery(G, s), asymptotically in terms of n? *(6 Points)*

Solution Task 7

Task 8: Minimum Spanning Trees *(20 Points)*

Let $G = (V, E)$ be a connected, undirected graph given via adjacency lists. Assume that G contains exactly one cycle.

(a) Describe an algorithm with runtime $O(|V|)$ that outputs the cycle in G. The output should have the form:

$$v_1, v_2, \ldots, v_k, v_1$$

where $v_1, v_2, \ldots, v_k$ are pairwise distinct nodes and each edge $\{v_i, v_{i+1}\}$ belongs to E for $i = 1, \ldots, k-1$, and $\{v_1, v_k\}$ belongs to E. Justify the runtime. *(8 Points)*

(b) Let $G = (V, E, w)$ be a connected, weighted graph and C a cycle in G. Let $e \in C$ be the heaviest edge in the cycle, i.e., $w(e) \geq w(e')$ for all edges $e' \in C$. Prove that there exists a minimum spanning tree (MST) of G that does not contain e. *(8 Points)*

(c) Describe an algorithm with runtime $O(|V|)$ that computes a minimum spanning tree of G. *(4 Points)*

Hint: You may use the results of (a) and (b) even if you did not solve them. If helpful, assume an adjacency matrix (with edge weights as entries) is also provided.

Solution Task 8