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

Algorithms and Data Structures Exam

25. Februar 2021, 9:00 -12:00

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.
- This is an **open book exam** therefore printed or hand-written material is allowed.
- **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.
- There is a **separate solution page** for each exercise.
- Write your name on **all sheets**!

Task	1	2	3	4	5	6	7	8	**Total**
Maximum	21	17	18	15	10	14	16	9	120
Points									

Task 1: Short Questions *(21 Points)*

(a) Let $h(x, i) := x + i \mod 7$ be a hash function with linear probing for collision resolution. Insert the keys 44, 45, 79, 55, 91, and 18 into the table using h. (3 Points)

0	1	2	3	4	5	6

(b) Provide a weighted, undirected graph G with positive weights and mark a node v and a shortest-path tree from v in G such that:

- G is not a tree,
- the minimum spanning tree of G is unique, and
- your marked shortest-path tree corresponds to the minimum spanning tree of G. (3 Points)

(c) Given two arrays A and B with $|A| = m$ and $|B| = n$ and $m \leq n$. The entries of the arrays are natural numbers. We want to determine if there is a number present in both A and B. Provide an efficient algorithm for this problem:

(i) assuming that finding and inserting into a hash table takes $O(1)$ time, as long as the load of the hash table is $O(1)$. (4 Points)

(ii) without using hashing. (5 Points)

Analyze the runtime for each case as a function of m and n.

(d) Given two red-black trees T_1 and T_2 with the same black height h. Additionally, all keys in T_1 are strictly smaller than all keys in T_2. Provide an algorithm that merges T_1 and T_2 into a valid red-black tree containing all keys from T_1 and T_2 in $O(h)$ time. Explain the runtime and why the resulting tree is valid. (6 Points)

Solution Task 1

Task 2: O-Notation *(17 Points)*

(a) Consider the following pseudocode:

Algorithm 1 `myst-div`(n)

```
1: while n > 1 do
2:     n = n/3                ▷ This step costs one time unit
3: return n
```

Let $T(n)$ represent the runtime of the function `myst-div(n)`, defined as the number of divisions performed in line 2. Determine the exact runtime $T(n)$. Then, using the definition of Big-O notation (i.e., without limits), show that $T(n) \in O(\log_{100} n)$. (5 Points)

Hint: You may restrict the domain of `myst-div` to the set $\{3^k \mid k \in \mathbb{N}\}$.

(b) State whether the following statements are true or false. Prove or disprove each statement using the definition of Big-O notation or limit-based characterization:

(i) $2\sqrt{n} + \log(n) \in o(n)$ (4 Points)

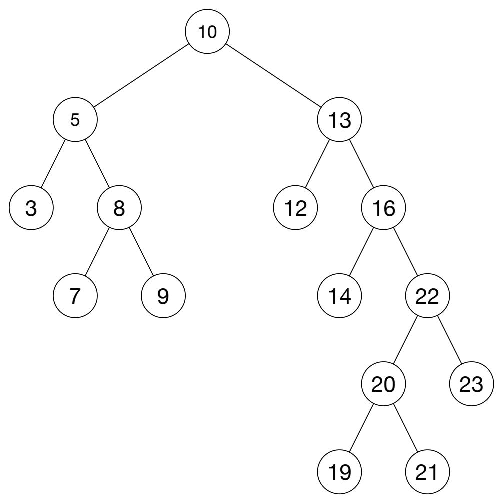
(ii) $2^{2n} \in \Theta(2^n)$ (3 Points)

(iii) $a^n \in \omega(n^k)$, for every real $a > 1$ and integer $k \geq 1$ (5 Points)

Solution Task 2

Task 3: Traversing Binary Search Trees *(18 Points)*

(a) Give the complete order in which the nodes are visited in pre-order, in-order, and post-order traversals.



(b) There exists only a single tree (ignoring the actual assigned numbers) such that pre-order, in-order and post-order traversals all result in the same order. Draw this tree and prove that for any other (non-empty) tree at least two of the different traversals result in a different order.

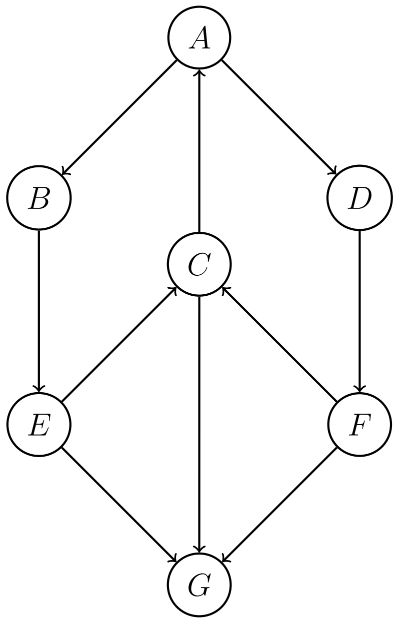
(c) Let T be a binary tree such that it holds that each node either has 2 children or it has no children at all (that is, either a node is a leaf or it has a left and a right child). Moreover, for each node u of T, we have stored a pair $(\mathsf{Pre}(u), \mathsf{Post}(u))$, where $\mathsf{Pre}(u)$ is the number of u in the pre-order traversal, and $\mathsf{Post}(u)$ is the number of u in the post-order traversal.

Give a constant-time algorithm that, given a node u of T, computes the size of the subtree rooted at u (the number of total nodes in T is not known). For example, in the above tree, the answer for the node with key value 16 should be 7, and the answer for the node with key value 7 should be 1.

Solution Task 3

Task 4: Graph Traversal *(15 Points)*

(a) Consider the graph below. Perform a depth first search starting at A. Label the edges as T, B, F and C, where T represents a tree edge, B a backward edge, F a forward edge, and C a cross edge. To guarantee the same solution, whenever deciding which node to pick, choose a node whose label occurs earliest in the alphabet. *(6 Points)*



(b) Suppose we have a connected undirected graph $G = (V, E)$ and a vertex $u \in V$. If we run DFS from u, we obtain a tree T. Suppose that if we run BFS from u, we obtain exactly the same tree T. Prove that $G = T$. *(9 Points)*

Solution Task 4

Task 5: Shortest Paths *(10 Points)*

(a) Construct a weighted directed graph G and highlight some node v of G, such that:

- G is not a tree,
- there is at least one edge of G of negative weight,
- Bellman-Ford would not detect negative cycles, and
- by running Dijkstra on G starting from v, the result *is* the shortest path tree of v.

(b) Construct a weighted directed graph G and highlight some node v of G, such that:

- G is not a tree,
- there is at least one edge of G of negative weight,
- Bellman-Ford would not detect negative cycles, and
- by running Dijkstra on G starting from v, the result *is not* the shortest path tree of v. Highlight the node for which the computed distance is wrong, say what is the right distance for that node, and what is the wrong distance obtained by running Dijkstra.

Solution Task 5

Task 6: Mystical Algorithm *(14 Points)*

Consider the following algorithm, which takes as input an array A of length n and a natural number m, where the entries in A are natural numbers between 0 and $m-1$.

Algorithm 2 `myst`

1: $B = [0] \cdot m$ ▷ create an array of length m where all entries are 0.
2: **for** $i = 0$ **to** $n-1$ **do**
3: $\quad B[A[i]] = B[A[i]] + 1$
4: $\ell = 0$
5: **for** $j = 0$ **to** $m-1$ **do**
6: $\quad$ **if** $B[j] > 0$ **then**
7: $\quad\quad$ **for** $k = 0$ **to** $B[j]-1$ **do**
8: $\quad\quad\quad A[\ell + k] = j$
9: $\quad\quad \ell = \ell + B[j]$

(a) What does the value $B[j]$ (after line 3) represent for $j \in \{0, \ldots, m-1\}$? (3 Points)

(b) Describe in one sentence what the algorithm `myst` does. (3 Points)

(c) Provide the asymptotic runtime of `myst` as a function of n and m, and justify your answer. (4 Points)

(d) Describe the advantages and disadvantages of `myst` compared to other algorithms you know that perform the same task. (4 Points)

Solution Task 6

Task 7: Dynamic Programming *(16 Points)*

Given a sequence of integers $S = (s_1, \ldots, s_n)$:

(a) Provide an algorithm (based on the principle of dynamic programming) that outputs the length of the longest increasing subsequence in S in $O(n^2)$ time. That is, the length k of the longest subsequence $(s_{i_1}, \ldots, s_{i_k})$ such that $s_{i_1} \leq \cdots \leq s_{i_k}$ and $i_1 < \cdots < i_k$. Justify the runtime. (10 Points)

Example: For $S = (3, 6, 9, 4, 2, 1, 5, 7, 8)$, one longest increasing subsequence is $(3, 4, 5, 7, 8)$, so the correct output for S is 5.

(b) Provide an algorithm that outputs the length of the longest bitonic subsequence in S in $O(n^2)$ time. That is, the length k of the longest subsequence $(s_{j_1}, \ldots, s_{j_k})$ such that $j_1 < \cdots < j_k$ and for some $\ell \in \{1, \ldots, k\}$, $s_{j_1} \leq \cdots \leq s_{j_\ell} \geq \cdots \geq s_{j_k}$. Justify the runtime. (6 Points)

Example: For $S = (3, 6, 9, 4, 2, 1, 5, 7, 8)$, one longest bitonic subsequence is $(3, 6, 9, 4, 2, 1)$, so the correct output for S is 6.

Solution Task 7

Task 8: Rabin-Karp Algorithm *(9 Points)*

Given the text $T = 334710367$ and the pattern $P = 103$ with a length of $m = 3$, both represented in base $b = 10$. The hash function uses the modulus $M = 11$. Let

$$t_s := T[s \ldots (s+m-1)] \mod 11$$

be the hash value of a substring of T used in iteration s of the Rabin-Karp algorithm.

(a) Fill in the missing values for t_s in the given table. (4 Points)

(b) Mark an $\times$ where the hash values of P and $T[s \ldots (s+m-1)]$ match in iteration s. Additionally, mark where P is identified as a match in T in iteration s. (5 Points)

Iteration s	0	1	2	3	4	5	6
Hash value t_s	4						
Hash match?	$\times$						
Pattern match?							

Solution Task 8