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

Algorithms and Data Structures Exam

10. März 2023, 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	**Total**
Maximum	25	15	20	15	15	10	20	120
Points								

Task 1: Short Questions *(25 Points)*

(a) Let $A = [1, 2, 3, 4, \ldots, n-2, n-1, 0]$ be an array of n elements, where all elements except the last one are already sorted. Provide the asymptotic runtime of Selection Sort and Insertion Sort on this array A. Assume that both algorithms behave exactly as described in the lecture. Justify your answers. *(5 Points)*

(b) Let A be an array of n numbers. In every such array, there exists an element p such that at most 50% of the elements are greater than and at most 50% of the elements are smaller than p. Assume we have an algorithm X that computes p in time $g(n)$.

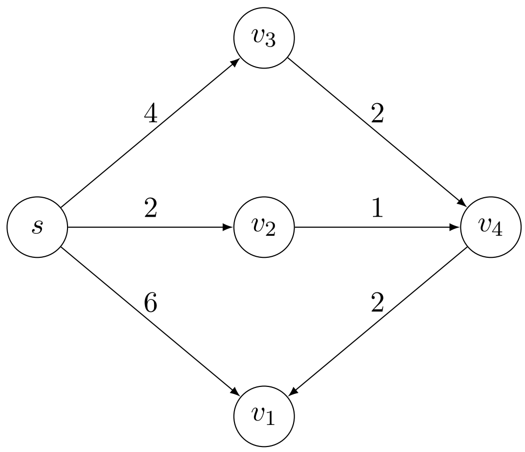
Now, we use a modified Quicksort algorithm that selects p (computed by X) as the pivot element. Apart from choosing the pivot differently, the algorithm functions exactly like the classical Quicksort from the lecture.

Provide a recurrence relation $T(n)$ that describes the runtime of this modified Quicksort. The recurrence should depend on $g(n)$ and does not need to be simplified. *(4 Points)*

(c) Insert the following sequence of keys into an initially empty binary search tree: $10, 5, 8, 15, 9, 3, 12, 7, 14, 18$.

Draw the resulting search tree. Then, show the tree after performing delete(9) followed by delete(10). *(6 Points)*

(d) Given the following directed graph with weighted edges and a start node s, determine the order in which Dijkstra's Algorithm processes the nodes (i.e., the order in which they are removed from the priority queue). *(3 Points)*



(e) To execute the Knuth-Morris-Pratt Algorithm (KMP) from the lecture, a prefix table S needs to be computed. Provide the array S for the pattern $P =$ "KKMLKMKKMM". *(7 Points)*

Task 2: Landau Notation *(15 Points)*

(a) Given the following five functions in terms of $n \in \mathbb{N}$, provide an ordering of the functions with respect to Big-O notation. That is, for two consecutive functions $f(n)$ and $g(n)$ in this ordering, it should hold that $f(n) \in O(g(n))$. You do not need to provide proofs. *(5 Points)*

- $a(n) = 5n^3 - 12n^2$
- $b(n) = 2^{\log_2(\log_2(n^2))}$
- $c(n) = \log_2(n) \cdot \left\lceil \frac{77}{2}n^2 + \frac{5}{2}n + 1 \right\rceil$
- $d(n) = \sqrt{n^4}$
- $e(n) = \log_2^2(n)$

Hint: The ceiling function $\lceil x \rceil$ is defined as $\lceil x \rceil := \min\{k \in \mathbb{Z} \mid k \geq x\}$.

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

$$\sum_{i=0}^{n} \frac{n \cdot i}{5} \in \Theta(n^3)$$

(4 Points)

(c) Prove the following statement using the definition of Big-O notation:

$$2^{\sqrt{\log_2(n)}} \in O(\sqrt{n})$$

(6 Points)

Task 3: Hashing *(20 Points)*

(a) Given the hash function:

$$h(x) = 3 \cdot x \mod m$$

where x is a key. Let $m = 11$, and insert the following keys into an initially empty hash table of size m, using linear probing:

$$3, 33, 17, 14, 25, 18$$

Provide the final state of the hash table after inserting all keys. *(4 Points)*

0	1	2	3	4	5	6	7	8	9	10

(b) Given three arrays A, B, and C, each containing exactly n elements, provide the most efficient algorithm to find three indices $0 \leq a, b, c < n$ such that:

$$A[a] + B[b] + C[c] = 0$$

Assume that such indices always exist. Also, provide the runtime complexity of your algorithm. *(10 Points)*

Hint: You may assume that insertion, lookup, and deletion in a hash table each take $O(1)$ time.

(c) Let $m > 1$ be an integer. Consider the family of hash functions:

$$H_m = \{h : x \mapsto (x + b) \mod m \mid b \in \{0, 1, \ldots, m - 1\}\}$$

Assume that the keys x are drawn from the set $\{0, 1, \ldots, 4m - 1\}$. Is this family of hash functions 1-universal? Prove or disprove. *(6 Points)*

Task 4: Mystical Algorithm *(15 Points)*

Consider the following pseudocode:

Algorithm 1 `Mystery`$A[0, \ldots, n-1]$

```
B = [ ]                                   ▷ empty array of length n
for i = 0 to n − 1 do
    s = 0
    for j = 0 to i do
        s = s + A[j]
    B[i] = s
return B
```

(a) Describe what the Mystery algorithm computes. Assume that A only stores integers. *(6 Points)*

(b) Provide the asymptotic runtime of Mystery, and justify your answer. *(3 Points)*

(c) Provide an optimized algorithm that computes the same result as Mystery, but with a better asymptotic runtime. Justify the runtime of your improved algorithm. *(6 Points)*

Task 5: Minimum number of squares *(15 Points)*

Given a natural number $n > 0$, the Minimal Number of Squares (MNS) Problem requires finding the smallest $k > 0$ such that there exist positive natural numbers $x_1, \ldots, x_k$ satisfying:

$$n = x_1^2 + x_2^2 + \cdots + x_k^2.$$

Provide an algorithm that solves the MNS problem in $O(n \cdot \sqrt{n})$ time. Justify your runtime analysis.
Example: For $n = 100$, the solution is $k = 1$ since $100 = 10^2$. For $n = 22$, the solution is $k = 3$ since $22 = 4 + 9 + 9$. The representation $22 = 16 + 4 + 1 + 1$ is correct but not minimal.
Hint: This problem can be solved using dynamic programming.

Task 6: Bipartite Graphs? *(10 Points)*

An undirected graph $G = (V, E)$ is called bipartite if the vertices can be partitioned into two disjoint sets A and B such that every edge connects a vertex from A to a vertex from B. Formally:

$$\forall \{a, b\} \in E : a \in A \text{ and } b \in B.$$

(Equivalently, no edge exists between two vertices within A, and no edge exists between two vertices within B.)

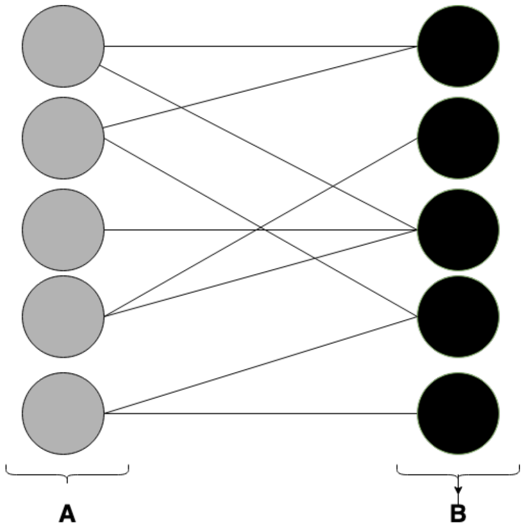


Figure 1: An example of a bipartite graph, the nodes in A are gray and the nodes in B are black.

Provide an algorithm that decides whether a given graph $G = (V, E)$ is bipartite in $O(m + n)$ time, where $n = |V|$ and $m = |E|$.
Assume that the graph G is given as an adjacency list and is connected.
Justify why your algorithm is correct and prove that it runs in the required time.

Task 7: Unique Minimum Spanning Trees *(20 Points)*

Given a connected, undirected graph $G = (V, E, w)$ with edge weights $w : E \to \mathbb{N}$. Define $n = |V|$ and $m = |E|$. Let T^* be a minimum spanning tree (MST) of G.

(a) Assume we increase the weight of an edge $e \in T^*$ by 1, while keeping all other weights unchanged. This results in a new weighted graph $G' = (V, E, w_e)$, where:

$$w_e(e) = w(e) + 1, \quad \text{and} \quad w_e(x) = w(x) \quad \forall x \neq e.$$

Let T' be an MST of G'. Prove that if:

$$w_e(T') = w(T^*)$$

then $T' \neq T^*$, i.e., the new MST must be different from the original one. *(6 Points)*

(b) Under the same conditions as in part (a), prove that if:

$$w_e(T') > w(T^*)$$

then the edge e must be present in every MST of G. *(10 Points)*

(c) A minimum spanning tree T^* is called unique if every other spanning tree T' has a greater weight than T^*.

Provide an algorithm that determines whether G has a unique MST. The algorithm takes as input a connected, undirected graph $G = (V, E, w)$ with weight function w.

Prove the correctness of your algorithm and show that it runs in $O(n \cdot m \cdot \log n)$ time. *(4 Points)*

Hint 1: Assume that the graph is given as an adjacency list. *Hint 2*: If you could not solve part (a) or (b), you may still use their results in this task.