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

Algorithm Theory

Monday, September 3, 2018, 10:00-12:00

Name: . . . .

Matriculation No.: . . . .

Signature: . . . .

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

- Put your **student ID** on the table next to you so we can check it.
- Write your **name** and **matriculation number** on this page and **sign** the document.
- Your **signature** confirms that you have answered all exam questions without any help, and that you have notified exam supervision of any interference.
- You are allowed to use a summary of **five (single-sided) A4 pages**.
- 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.
- **No electronic devices** are allowed.
- 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.
- A total of 40% of all possible points (**48 points**) is sufficient to pass this exam.
- A total of 80% of all possible points (**96 points**) is sufficient for the best grade.
- There is a **separate solution page** for each exercise and two additional **blank pages at the end**.
- Write your name on **all sheets**!

Task	1	2	3	4	5	6	**Total**
Maximum	43	12	14	13	12	26	120
Points							

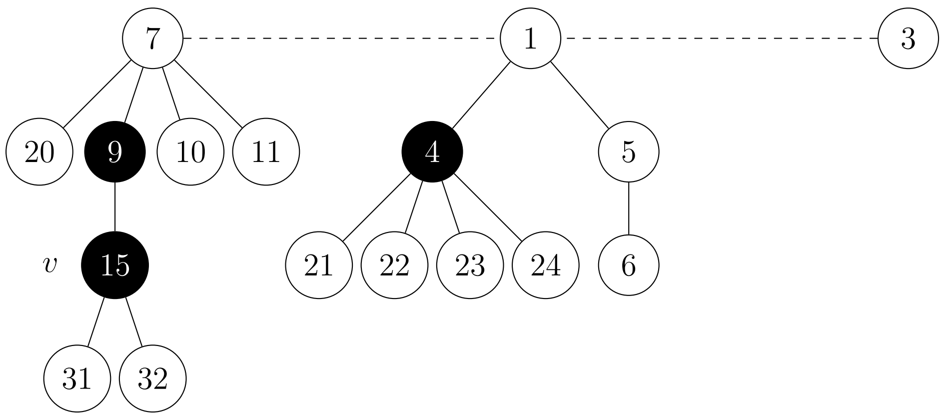
Task 1: Short Questions *(43 Points)*

(a) For the following two statements, state whether they are true or false, and explain why.

(I) ***(6 Points)*** Let $G = (V, E)$ be a connected, undirected graph with edge-weight function $w : E \to \mathbb{R}$ and assume that all edge weights are distinct. Consider a cycle $(v_1, v_2, \ldots, v_{k+1})$ of length k, where $\{v_j, v_{j+1}\} \in E$ for all $j \leq k$, and $v_1 = v_{k+1}$. Let $\{v_i, v_{i+1}\}$ be the edge in the cycle with the largest edge weight. Then, edge $\{v_i, v_{i+1}\}$ does not belong to the minimum spanning tree T of G.

(II) ***(5 Points)*** Assume that we have a counter C that is represented by n binary bits. The counter supports two operations: `increment` and `decrement`. The cost for each operation is the number of bits it flips to represent the new number. Then the worst-case amortized cost per operation for an arbitrary sequence of operations is $\Theta(n)$.

(b) ***(7 Points)*** Consider the following Fibonacci heap (black nodes are marked, white nodes are unmarked). How does the given Fibonacci heap look after a `decrease-key`$(v, 2)$ operation and how does it look after a subsequent `delete-min` operation?



(c) ***(6 Points)*** Let D be a data structure that stores integers and supports two operations; `Insert`(X) inserts integer X into D and `Remove-Min` removes and returns the smallest integer in D. When there are n elements in D, the cost of an `Insert` operation is T_n and the cost of a `Remove-Min` is $O(\sqrt{\log n})$. Let us assume that the fastest algorithm to sort n integers runs in time $\Omega(n \log n)$. Prove that $T_n \in \Omega(\log n)$.

(d) ***(6 Points)*** Let A be an unsorted array of n distinct integers. Describe an algorithm that finds the k^{th} smallest integer in $O(n)$ *expected* time.

Remark: No need to do the runtime analysis!

(e) ***(8 Points)*** Consider the PRAM model with $\lceil n/2 \rceil$ processors. Assume that memory cells $c_1, \ldots, c_n$ contain integers. Describe a parallel EREW algorithm that computes the maximum integer in $c_1, \ldots, c_n$ in $O(\log n)$ depth (number of parallel steps). Explain the correctness and running time (depth) of your algorithm.

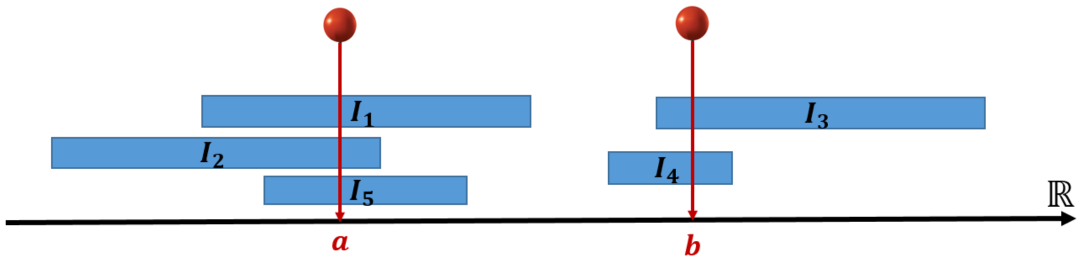
(f) ***(5 Points)*** Consider an undirected graph $G = (V, E)$ with $2n$ nodes, i.e., $|V| = 2n$. Prove that G has matching of size $n/2$ if every node $v \in V$ has degree n.

Solution Task 1

Task 2: Stabbing Intervals *(12 Points)*

Given a set I of n intervals on the real line, a set of real numbers S is said to stab I if for every $I' \in I$, there is a real number $s \in S$ such that $s \in I'$.

As an example $S = \{a, b\}$ stabs $I = \{I_1, I_2, I_3, I_4, I_5\}$ in the following picture.



Provide an efficient greedy algorithm that for any given set I of intervals on the real line, finds a minimum cardinality set of real numbers that stabs I. Prove the correctness of your algorithm.

Solution Task 2

Task 3: Balanced Partitioning *(14 Points)*

You are given a set $\mathcal{X}$ of n integers in the range of $[0, K]$. The goal is to partition S into two subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ such that $|S_1 - S_2|$ is minimized, where S_1 is the sum of the integers in $\mathcal{X}_1$ and S_2 is the sum of the integers in $\mathcal{X}_2$.

Provide an algorithm to achieve the goal in time $O(n^2K)$. Explain why the running time of your algorithm is $O(n^2K)$.

Solution Task 3

Task 4: Amortized Analysis *(13 Points)*

You are given a data structure which consists of a *singly linked list* that offers two operations. The first operation is `Insert`(x), which inserts an element with a unique integer key x in front of the list as the new head. The second operation is `Split`(x), which splits off and discards the element with key x and all elements in front of it, so that afterwards the list starts with the successor of x as a new head. Assume that all inserted keys are *unique* and that whenever `Split`(x) is called, the key x *is contained* in the current list.

(a) ***(4 Points)*** Briefly explain how these two operations can be implemented. Give the running time using the O-Notation. Assume that the list contains n elements and that running time is measured by the number of pointers that have to be read.

(b) ***(9 Points)*** Define a suitable *potential function* to prove that a series of n `Insert` and `Split` operations has an amortized running time of $O(1)$ per operation (still under the assumption that x is contained in the list if `Split`(x) is called).

Solution Task 4

Task 5: Approximation Algorithms *(12 Points)*

Let $G = (U \dot\cup V, E)$ be a bipartite graph. We say that G is *light* if all the nodes in U have degree exactly 2. We call a subset $R \subseteq V$ a representative of G if every node in U has a neighbor in R. We consider the problem of finding a minimum size representative in a given light bipartite graph.

Provide an efficient 2-approximation algorithm to solve the problem. Explain why your algorithm is a 2-approximation.

Hint: One possible solution is to reduce the problem to a known problem from the lecture. Try to first construct a graph H, which is defined only on the vertices in V.

Solution Task 5

Task 6: Randomized and Online Algorithms *(26 Points)*

Let $G = (V, E)$ be an arbitrary unweighted undirected graph. A maximum cut of G is a cut whose size is at least the size of any other cut in G.

(a) ***(4 Points)*** Give a simple randomized algorithm that returns a cut of size at least $1/2$ times the size of a maximum cut *in expectation* and prove this property.

(b) ***(10 Points)*** Prove that the following deterministic algorithm (Algorithm 1) returns a cut of size at least $1/2$ times the size of a maximum cut.

Algorithm 1 Deterministic Approximate Maximum Cut

```
Pick arbitrary nodes v_1, v_2 ∈ V
A ← {v_1}
B ← {v_2}
for v ∈ V \ {v_1, v_2} do
    if deg_A(v) > deg_B(v) then        ▷ deg_X(v) is the number of v's neighbors in X ⊆ V.
        B ← B ∪ {v}
    else
        A ← A ∪ {v}
Output A and B
```

(c) ***(5 Points)*** Let us now consider an online version of the maximum cut problem, where the nodes V of a graph $G = (V, E)$ arrive in an online fashion. The algorithm should partition the nodes V into two sets A and B such that the cut induced by this partition is as large as possible. Whenever a new node $v \in V$ arrives together with the edges to the already present nodes, an online algorithm has to assign v to either A or B. Based on the above deterministic algorithm (Alg. 1), describe a deterministic online maximum cut algorithm with *strict competitive ratio* at least $1/2$. You can use that fact that Algorithm 1 computes a cut of size at least half the size of a maximum cut.

Hint: An online algorithm for a maximization problem is said to have strict competitive ratio α if it guarantees that $\text{ALG} \geq \alpha \cdot \text{OPT}$, where ALG and OPT are the solutions of the online algorithm and of an optimal offline algorithm, respectively.

(d) ***(7 Points)*** Show that no deterministic online algorithm for the online maximum cut problem can have a strict competitive ratio that is better than $1/2$.

Solution Task 6

Additional Sheet

Additional Sheet