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

Algorithm Theory

September 4, 2020, 13:00-15:00

Name: ....

Matriculation No.: ....

Signature: ....

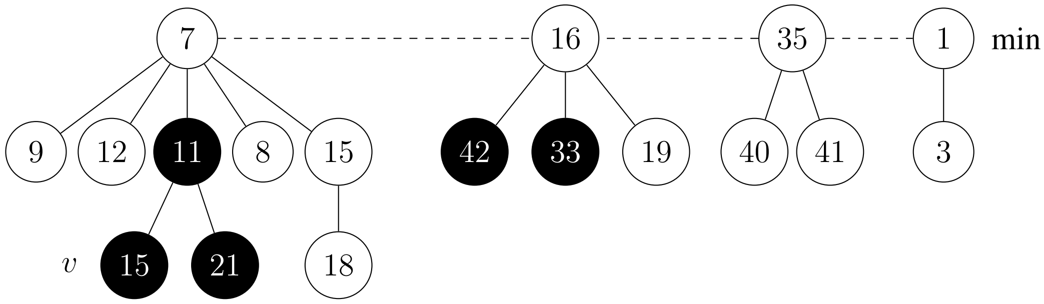
Do not open or turn until told so by the supervisor!

- Write your **name** and **matriculation number** on this page and **sign** the document.
- Your **signature** confirms that you have answered all exam questions 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.
- Write your name on **all sheets**!

Task	1	2	3	4	5	**Total**
Maximum	40	28	15	15	22	120
Points						

Task 1: Short Questions *(40 Points)*

(a) *(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?



(b) *(6 Points)* Let $G = (V, E)$ be a flow network with source s and sink t and non-negative integer capacities. Let $(S, V \setminus S)$ be a minimum s-t cut with capacity C and e a crossing edge of the cut, i.e., an edge going from S to $V \setminus S$. Describe how to decide in time $O(|E| \cdot C)$ whether or not e is a crossing edge of *all* minimum s-t cuts.

(c) *(6 Points)* Consider a game board with r rows and c columns (i.e., an $r \times c$ grid). Imagine that a robot sits on the upper left cell of this board. The robot can only move in two directions; to the neighboring cell on the right or the neighboring cell below. On each cell (i, j), there is a number $n_{(i,j)}$ of coins that the robot picks up when stepping on that cell. Describe an $\mathcal{O}(r \cdot c)$-time algorithm to find a path for the robot from the top-left to the bottom-right cell of the board such that the total number of coins that the robot collects is maximized. Analyze the running time.

(d) *(8 Points)* Consider the EREW PRAM model. Assume we are given an array A of n positive integers and a positive integer x which is stored in the shared memory. We want to find the number of subarrays of A whose entries sum up to x. That is, we want to find the number of pairs (i, j) with $0 \leq i < j \leq n - 1$ such that $\sum_{k=i}^{j} A[k] = x$. Show that there is an algorithm that solves this problem in time $\mathcal{O}(\log^2 n)$ using n processors.

Hint: Solve the problem in the CREW PRAM model first.

(e) *(7 Points)* A *hypergraph* is a generalization of an undirected graph in which edges consist of arbitrary subsets of vertices. That is, a hypergraph consists of a set of nodes V and a set of edges $E \subseteq \mathcal{P}(V) \setminus \emptyset$. A hypergraph is called k-uniform if each edge has size k, i.e., contains exactly k nodes. For example, a simple, undirected graph is a 2-uniform hypergraph.

A matching of a hypergraph is a set of edges which are pairwise disjoint. The aim is to approximate a maximum matching of a 3-uniform hypergraph.

(i) Provide a 3-uniform hypergraph G and a maximal matching M such that $|M| = \frac{|M^*|}{3}$ where M^* is a maximum matching.

(ii) Show that any maximal matching of a 3-uniform hypergraph is a $(1/3)$-approximate solution of a maximum matching.

(f) *(6 Points)* Consider a counter C represented by an n-bit binary string which is initially set to 0. On C we can execute the operations `increment` and `decrement`, which increment/decrement C by 1. The cost of an operation is the number of bits that are flipped.

What can you say about the worst-case amortized cost per operation if you only have `increment` operations compared to the case where we allow an arbitrary sequence of `increment` *and* `decrement` operations?

Solution Task 1

Task 2: Vertex Coloring *(28 Points)*

A vertex coloring of a graph is an assignment of colors to the vertices such that adjacent nodes have different colors. Assume we are given a simple, undirected graph $G = (V, E)$ with n vertices such that G can be colored with only 3 colors. We want to show that there is a polynomial-time algorithm to color G with at most $O(\sqrt{n})$ colors.

(a) *(4 Points)* Show that for every vertex $u \in V$, the graph induced by the neighbors of u is a bipartite graph. That is, (V', E') is a bipartite graph where

$$V' = \big\{v \in V \big| \{u, v\} \in E\big\} \text{ and } E' = \big\{\{v, w\} \in E \big| v, w \in V'\big\}$$

(b) *(6 Points)* Show that a bipartite graph with n nodes and m edges can be colored with 2 colors in time $\mathcal{O}(m + n)$.

(c) *(7 Points)* Describe how to color a part of the nodes with at most $O(\sqrt{n})$ different colors in polynomial time, such that afterwards, there is no uncolored node which has more than $\sqrt{n}$ uncolored neighbors.

Hint: Look for an uncolored node u with more than $\sqrt{n}$ uncolored neighbors and use parts (a) and (b). How often do you need to repeat this process?

(d) *(6 Points)* Show that every graph with maximum degree Δ can be colored with $\Delta + 1$ colors in polynomial time.

(e) *(5 Points)* Conclude the proof by showing that there is an algorithm to color G with at most $O(\sqrt{n})$ colors in polynomial time.

Hint: You can use parts (c) and (d) also if you did not solve them.

Solution Task 2

Task 3: Online Algorithms *(15 Points)*

Assume you have 1 Euro and want to exchange it to Dollars during the next k days. More specifically, you must choose a day $i \in \{1, \ldots, k\}$ on which you trade the whole Euro. Exchanging back and forth or partly exchanging the Euro is not allowed. On each day i you learn the exchange rate x_i which is valid for that day (i.e., x_i is the amount of Dollars you get for 1 Euro). We assume $x_i \in [1, a]$, where $a \geq 1$ is a real number which is known. The aim is to get the maximum amount of Dollars for your 1 Euro.

(a) *(8 Points)* Give a deterministic online algorithm such that $\text{ALG} \geq \frac{\text{OPT}}{\sqrt{a}}$, where ALG is the amount of Dollars given by your algorithm and OPT the amount given by an optimal solution. Prove the competitive ratio.

(b) *(7 Points)* Prove that there is no deterministic algorithm with a better competitive ratio than the one from part (a). That is, for $c < \sqrt{a}$ there is no deterministic algorithm such that $\text{ALG} \geq \frac{\text{OPT}}{c}$ in all cases.

Solution Task 3

Task 4: Presentation Scheduling *(15 Points)*

Assume there are n students $s_1, \dots, s_n$. Each student has finished some individual project and now has to present the results to some professors. There are k professors $p_1, \dots, p_k$. Each professor p_i hands in a list $L_i \subseteq \{s_1, \dots, s_n\}$ of students for whose projects he/she is an expert. Each professor p_i is willing to attend at most a_i presentations.

The exam regulations require that at each presentation, x professors that are experts on the topic are present.

(a) *(8 Points)* Describe a polynomial-time algorithm that computes an assignment of the professors to the student's presentations such that the given constraints are fulfilled, or reports that no such assignment exists.

(b) *(7 Points)* As there is a shortage of professors, the university loosens the requirements such that among the x professors that need to be present at each presentation, at least y need to be an expert on the topic, for some $y < x$. Describe how to construct a feasible schedule in this case.

Solution Task 4

Task 5: Randomization - Dominating Set in Regular Graphs *(22 Points)*

Let $G = (V, E)$ be an undirected graph. A set $D \subseteq V$ is called a *dominating set* if each node in V is either contained in D or adjacent to a node in D.

We consider the following randomized algorithm for d-regular graphs (i.e., graphs in which each node has exactly d neighbors).

Algorithm 1 `domset`(G)

1: $D = \emptyset$
2: Each node joins D independently with probability $p := \min\{1, \frac{c \cdot \ln n}{d+1}\}$ for some constant $c > 0$
3: Each node that is neither in D nor has a neighbor in D joins D
4: **return** D

For simplicity, in all tasks you may assume that $\frac{c \cdot \ln n}{d+1} \leq 1$, i.e., that $p = \frac{c \cdot \ln n}{d+1}$.

(a) *(6 Points)* Show that for $c \geq 1$, the expected size of D (after the execution of `domset`) is at most $\frac{cn \ln n}{d+1} + 1$.
Hint: Use the inequality $(1 - x) \leq e^{-x}$.

(b) *(6 Points)* Show that after line 2 of `domset`, D has size $O\left(\frac{n \ln n}{d+1}\right)$ with probability at least $1 - \frac{1}{n}$.
Hint: You might want to use Chernoff's Bound: If $X_1, \ldots, X_n$ *is a sequence of independent 0-1 random variables,* $X = \sum X_i$ *and* $\mu = E[X]$*, then for any* $\delta > 0$ *we have*

$$\Pr(X \geq (1 + \delta)\mu) \leq e^{-\frac{\min\{\delta, \delta^2\}}{3}\mu} .$$

(c) *(4 Points)* Show that for $c \geq 2$, with probability at least $1 - \frac{1}{n}$, no node joins D in line 3 of `domset`.

(d) *(2 Points)* Conclude that for $c \geq 2$, `domset` returns a dominating set of size $O\left(\frac{n \ln n}{d+1}\right)$ with probability at least $1 - \frac{2}{n}$.

(e) *(4 Points)* Show that for $c \geq 2$, `domset` computes an $\mathcal{O}(\ln n)$-approximation of a minimum dominating set (i.e., a dominating set of minimum size) with probability at least $1 - \frac{2}{n}$.
Hint: You can use part (d) also if you did not solve it.

Solution Task 5