Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department Informatik
Johannes Lengler
Jingqiu Ding
Manuel Wiedmer

David Steurer
Hongjie Chen
Lucas Slot

Exam

Algorithmen und Datenstrukturen

August 16, 2024

DO NOT OPEN!

Student number: _____

Seat number: _____

Good luck!

	T1 (26P)	T2 (14P)	T3 (8P)	T4 (12P)	Prog. (40P)	**Σ** (100P)
Score						
Corrected by						

Theory Task T1. **/ 26 P**

In this problem, you have to provide **solutions only**. You do not need to justify your answer.

/ 6 P a) *Asymptotic notation quiz:* For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. You get at least 0 points in total.

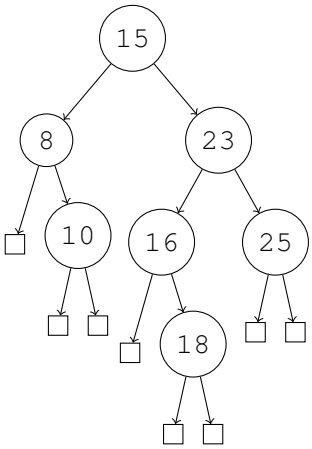
Assume $n \geq 4$.

Claim	true	false
$n^{3.5} - n^{2.5} \geq \Omega(n^3)$	☐	☐
$\frac{n^3}{\log(n)} = \Theta(n^3)$	☐	☐
$2^{(n^2)} \leq O(3^{3n})$	☐	☐
$\sum_{i=1}^{\lceil\sqrt{n}\rceil} \sqrt{i} \geq \Omega(n)$	☐	☐
Suppose $a_n < b_n$ for all $n \in \mathbb{N}$ and $a_n \geq \Omega(n)$. It follows that $b_n - a_n \geq \Omega(n)$.	☐	☐
$1 + n + (-1)^n \cdot n = \Theta(n)$	☐	☐

/ 3 P b) *Binary trees:*

i) Draw the **binary search tree** with minimum possible depth containing the keys 1, 2, 3, 4, 5, 6, 7. The depth of a tree is the length of the longest path from the root to any of its leaves. If there are several solutions, it is enough to draw only one of them.

ii) Draw the **AVL tree** obtained from the following tree by performing the operations `INSERT`(11) and `DELETE`(25). It suffices to only draw the final tree. If there are several solutions, it is enough to draw only one of them.



iii) Draw the **max-heap** obtained from inserting the keys 2, 3, 11, 13, 5, 19 in this order into an empty heap. It suffices to only draw the final heap. If there are several solutions, it is enough to draw only one of them.

/ 5 P

c) *Graph quiz:* For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. You get at least 0 points in total.

As a reminder, here are a few definitions for an undirected graph $G = (V, E)$:

For $k \geq 2$, a *walk* is a sequence of vertices $v_1, \ldots, v_k$ such that for every two consecutive vertices v_i, v_{i+1}, we have $\{v_i, v_{i+1}\} \in E$.

A *closed walk* is a walk with $v_1 = v_k$.

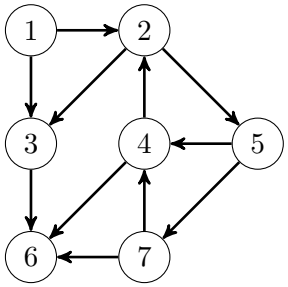
A *cycle* is a closed walk where $k \geq 3$ and all vertices (except v_1 and v_k) are distinct.

A *closed Eulerian walk* is a closed walk which traverses every edge in E exactly once.

A *spanning tree* of G is a subset $F \subseteq E$ such that (V, F) is a tree (in particular, connected).

Claim	**true**	**false**
Let $G = (V, E)$ be an undirected graph. If the degree of each vertex of G is at least two, then G is connected.	☐	☐
Let $G = (V, E)$ be a connected, undirected graph. Suppose there is an edge $e \in E$ such that G would remain connected after removing e from E. Then the original graph G contains a cycle.	☐	☐
There exists an undirected graph $G = (V, E)$ with 9 vertices such that each vertex has degree 5.	☐	☐
Let $G = (V, E)$ be a connected, undirected graph with $n \geq 10$ vertices. If G has exactly two vertices with odd degree, then we can remove an edge from E so that the resulting graph contains a closed Eulerian walk.	☐	☐
Let $G = (V, E)$ be a connected, undirected graph with $n \geq 10$ vertices. Then G has at most n^2 different spanning trees.	☐	☐

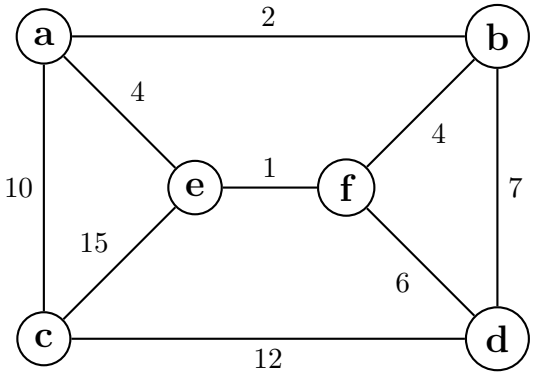
/ 2 P d) *Breadth/depth-first search:* Consider the following directed graph:



i) Draw the breadth-first tree resulting from a breadth-first search starting from vertex 1. Process the neighbors of a vertex in increasing order.

ii) Draw the depth-first tree resulting from a depth-first search starting from vertex 1. Process the neighbors of a vertex in increasing order.

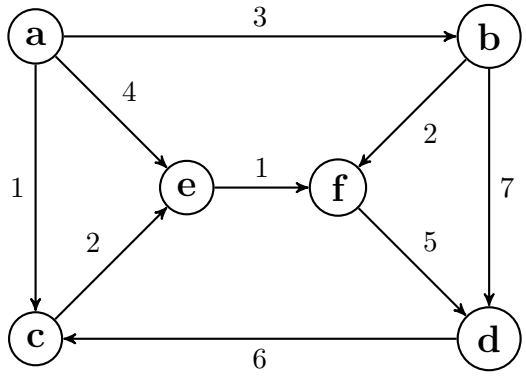
/ 2 P e) *Minimum Spanning Tree:* Consider the following graph:



i) Find a minimum spanning tree. You can either highlight its edges in the picture above, or draw the minimum spanning tree below. (If there are more than one minimum spanning trees, it suffices to find just one of them.)

ii) List all edges that are not part of any minimum spanning tree.

/ 2 P f) *Shortest Path Tree:* Consider the following graph:



i) Find the shortest-path tree rooted at vertex a (i.e., the output of Dijkstra's algorithm if we were to start from vertex a). You can either highlight its edges in the picture above, or draw the shortest-path tree below.

ii) Does the above graph have a topological ordering? If yes, write down one topological ordering. If no, give an argument.

/ 6 P

g) *Sorting and searching algorithms:* For parts i) and ii), there is only one correct choice. Please circle the correct answer for each question. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. For the whole part g), together with iii) and iv) below, you get at least 0 points.

i) There exist arrays of length n for which `MergeSort` needs time $\Omega(n^2)$.

(a) True

(b) False

ii) On the array $[n, n-1, \ldots, 1]$, the runtime of `QuickSort` (to sort the array in ascending order) is $O(n \log(n))$.

(a) True, for all possible pivot choices.

(b) False, for all possible pivot choices.

(c) This depends on the pivot choices.

For parts iii) and iv) there is at least one correct choice, but there might be several ones. Please circle *all* correct answers for each question. For each subtask, if you circle nothing, you get 0 points. If you circle at least 1 option, then you get +0.5 points for every correctly answered (circled or not circled) option and -0.5 points for every wrongly answered option. For the whole part g), together with i) and ii) above, you get at least 0 points.

iii) Which of the following algorithms has runtime $O(n \log(n))$ on the array $[n, n-1, \ldots, 1]$ (to sort the array in ascending order)?

(a) `HeapSort`

(b) `BubbleSort`

(c) `InsertionSort`

(d) `SelectionSort`

iv) For every comparison based search algorithm working on sorted arrays, there exist arrays of length n, for which the runtime of the algorithm is

(a) $\Omega(\log(n))$

(b) $\Omega(n^{0.99})$

(c) $O(\log(n))$

(d) $\Theta(n)$

Theory Task T2. **/ 14 P**

In this part, you should **justify your answers briefly**.

/ 4 P a) *Counting iterations:* For the following code snippets, derive an asymptotic bound for the number of times f is called. Simplify the bound as much as possible and state it in Θ-notation as concisely as possible.

i) Snippet 1:

Algorithm 1

for $i = 1, 2, 3, \ldots, n$ **do**
  for $k = 1, 2, 3, \ldots, i^2$ **do**
    $f()$

ii) Snippet 2:

Algorithm 2

$i = 1$
while $i \leq n$ **do**
  for $j = i, i+1, i+2, \ldots, 2i$ **do**
    $f()$
  $i \leftarrow i + 1$

/ 3 P b) *Induction:* Prove the following inequality by induction: for every integer $n \geq 1$,

$$\frac{n^3}{2} \geq \sum_{i=1}^{n} i(i-1)\,.$$

/ 4 P c) *Shortest path:* Decide whether the following statement is true. If yes, give a sketch of a proof; if not, provide a counterexample.

For any arbitrary undirected graph with positive edge weights, every edge in any shortest path tree is always contained in at least one of the minimum spanning trees.

/ 3 P

d) *Depth first search:* Given is a binary tree T and a root v with each of the n nodes being assigned an integer value. Design an algorithm that decides whether there is a root-to-leaf path such that the sum of the values of the nodes in the path is equal to a given integer k. The algorithm should run in time $O(n)$. Describe your algorithm in pseudocode and briefly justify its runtime.

/ 8 P

Theory Task T3.

In this problem, you are given an array $A = [a_1, \ldots, a_n]$ of n positive integers, i.e. $a_i \in \mathbb{Z}_{\geq 1}$. Recall that $a_{i_1}, \ldots, a_{i_k}$ is a subsequence of the array A if the indices satisfy $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. A subsequence $a_{i_1}, \ldots, a_{i_k}$ of A is called a Fibonacci subsequence if every element in the subsequence is the sum of the previous two elements, i.e. $a_{i_j} = a_{i_{j-1}} + a_{i_{j-2}}$ for all $3 \leq j \leq n$.

Provide a *dynamic programming* algorithm that, given such an array A as above, returns the length of a longest Fibonacci subsequence (but not the subsequence itself). For example,

- For the array $A = [1, 1, 2, 4, 5, 3, 9]$, the output should be 4 and the corresponding subsequence is $1, 1, 2, 3$ or $1, 4, 5, 9$.
- For the array $A = [5, 2, 1, 3, 4, 10, 7]$, the output should be 5 and the corresponding subsequence is $2, 1, 3, 4, 7$.
- For the array $A = [1, 2, 1, 4, 1, 1, 10]$, the output should be 2 and the corresponding subsequence is (for example) $1, 1$.

In order to obtain full points, your algorithm should run in time $O(n^3)$. You can still get a maximum of 4 points for a correct solution that runs in time $O(n^4)$.

If you give a correct algorithm that runs in time $O(n^2 \log(n))$, then you can get 2 *bonus* points (i.e. $8+2 = 10$ points). Note: This solution is hard. We recommend to only try this after you have solved the rest of the exam. You can get full points (and a grade of 6) for the exam without these bonus points.

In your solution, address the following aspects:

1. *Dimensions of the DP table*: What are the dimensions of the DP table?
2. *Subproblems*: What is the meaning of each entry?
3. *Recursion*: How can an entry of the table be computed from previous entries? Specify the base cases of the recursion, i.e., the cases that do not depend on others.
4. *Calculation order*: In which order can entries be computed so that values needed for each entry have been determined in previous steps?
5. *Extracting the solution*: How can the solution be extracted once the table has been filled?
6. *Running time*: What is the running time of your solution?

Dimensions of the DP table:

Scheme continues on the next page.

Subproblems:

Recursion:

Calculation order:

Extracting the solution:

Running time:

Theory Task T4. **/ 12 P**

You are asked by the fire department of Zurich to design some algorithms to optimize their infrastructure. The city of Zurich consists of n *neighbourhoods* $N_1, N_2, \ldots, N_n$, and m **one-way** *roads* $R_1, R_2, \ldots, R_m$, which each connect two neighbourhoods. For $j \in [m]$, road R_j is $\ell_j \in \mathbb{N}$ kilometres long. A fire truck drives at a speed of 1 kilometre per minute (so it can cross road R_j in ℓ_j minutes). You can assume that any neighbourhood is connected to any other neighbourhood by a sequence of roads.

In this exercise, when asked to describe an algorithm, you must address the following elements:

1) The graph that you use. This includes: its vertex set and edge set; whether edges are directed or not; whether vertices or edges are weighted.
2) The algorithm that you apply to this graph.
3) How you extract the solution from (the output of) your algorithm.
4) The total running time of your proposed algorithm, with a short justification.

You can use the algorithms covered in the lecture material as subroutines, and you can use their running time bounds without proof.

Exercise continues on the next page.

/ 4 P

a) The fire department wants to build a fire station in one of the neighbourhoods. Let $T_{\max} \in \mathbb{N}$. A fire truck should be able to reach any other neighbourhood in Zurich from the station in at most $T_{\max}$ minutes. Describe an algorithm which outputs a suitable neighbourhood to build the station in if it exists, and outputs IMPOSSIBLE otherwise. The runtime of your algorithm should be at most $O\big(n \cdot (m+n) \cdot \log n\big)$.

Graph:

Algorithm:

Extraction:

Runtime:

Exercise continues on the next page.

/ 4 P

b) The fire department wants to build *two* water towers and connect each neighbourhood to *at least* one of the towers by pipe (though not necessarily the same tower for each neighbourhood). Each tower must be placed in a neighbourhood, and the pipes can only be laid beneath already existing roads. Water can flow through a pipe **in both directions**. Let $L_{\max} \in \mathbb{N}$. Describe an algorithm which outputs POSSIBLE if the towers can be placed so that at most $L_{\max}$ kilometres of pipe is needed to connect all neighbourhoods to at least one tower, and outputs IMPOSSIBLE otherwise. The runtime of your algorithm should be at most $O(m \log m)$.

Graph:

Algorithm:

Extraction:

Runtime:

Exercise continues on the next page.

/ 4 P

c) Sometimes, a fire truck must respond to a call while it is not at the station. Let $T'_{\max} \in \mathbb{N}$. The fire department wants to know if it is possible for a fire truck to travel between any pair of neighbourhoods in at most $T'_{\max}$ minutes. To make this a bit easier, a fire truck is allowed to turn on its sirens **once per trip**, allowing it to drive **twice as fast** on a **single road**. The fire truck is allowed to use its sirens on a different road each trip. For instance, it might activate the sirens on road R_5 on its trip from N_1 to N_4, but activate its sirens on road R_3 on its trip from N_6 to N_2. Describe an algorithm which outputs POSSIBLE if a fire truck can travel between any pair of neighbourhoods in at most $T'_{\max}$ minutes (using its sirens at most once per trip), and outputs IMPOSSIBLE otherwise. To achieve full points, the runtime of your algorithm should be at most $O(n^3)$. You can still earn 3 points for a correct solution with runtime at most $O(m \cdot n^3)$.

Graph:

Algorithm:

Extraction:

Runtime:

Extra space for notes. If you write any answers here that should be graded, please indicate clearly to which task your notes belong, and make a note in the main body of the exam.

Extra space for notes. If you write any answers here that should be graded, please indicate clearly to which task your notes belong, and make a note in the main body of the exam.

Extra space for notes. If you write any answers here that should be graded, please indicate clearly to which task your notes belong, and make a note in the main body of the exam.

Extra space for notes. If you write any answers here that should be graded, please indicate clearly to which task your notes belong, and make a note in the main body of the exam.