Eidgenössische
Technische Hochschule
Zürich

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

Institut für Theoretische Informatik
Peter Widmayer
Tobias Pröger

Exam
Datenstrukturen und Algorithmen
D-INFK

August 7, 2014

Last name, first name: _____

Student number: _____

With my signature I confirm that I was able to participate in the exam under regular conditions, and that I read and understood the notes below.

Signature: _____

Please note:

- You may not use any accessories except for a dictionary and writing materials.
- Please write your student number on **every** sheet.
- **Immediately** report any circumstances that disturb you during the exam.
- Use a new sheet for every problem. You may only give one solution for each problem. Invalid attempts need to be clearly crossed out.
- Please write **legibly** with blue or black ink. We will only grade what we can read.
- You may use algorithms and data structures of the lecture without explaining them again. If you modify them, it suffices to explain your modifications.
- You have 180 minutes to solve the exam.

Good luck!

Student number: _____

problem	1	2	3	4	**Σ**
max. score	16	13	6	10	45
$\sum$ score					

Problem 1.

Please note:

1) In this problem, you have to provide **solutions only**. You can write them right on this sheet.
2) If you use algorithms and notation other than that of the lecture, you need to **briefly** explain them in such a way that the results can be understood and checked.
3) We assume letters to be ordered alphabetically and numbers to be ordered ascendingly, according to their values.

1 P a) Perform two iterations of *Insertion Sort* on the following array. The array has already been sorted by previous iterations up to the double bar.

2	5	7	15	9	11	8	3	1	12	14	20
1	2	3	4	5	6	7	8	9	10	11	12

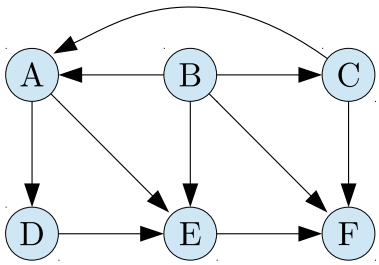
1	2	3	4	5	6	7	8	9	10	11	12

1	2	3	4	5	6	7	8	9	10	11	12

1 P b) Insert the keys 9, 11, 17, 25, 31, 20 in this order into the following hash table below. Use double hashing with the hash function $h(k) = k \bmod 11$, and use $h'(k) = 1 + (k \bmod 9)$ for probing.

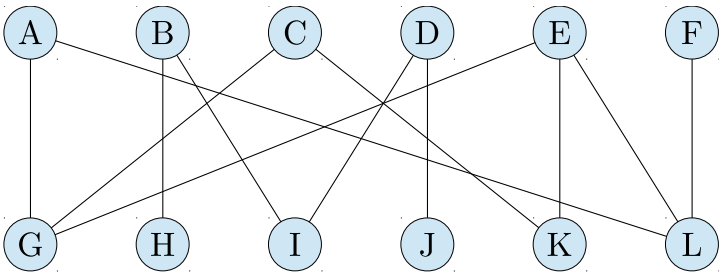
0	1	2	3	4	5	6	7	8	9	10

1 P c) Provide a topological ordering for the graph below.



Topological ordering: _____, _____, _____, _____, _____, _____

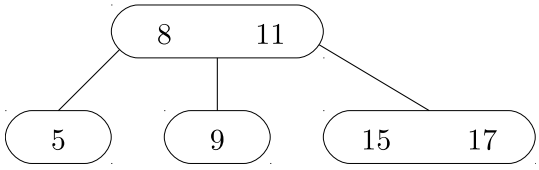
1 P d) Provide a subset of vertices of the following graph that is *as small as possible* and that proves using Hall's theorem that the graph has no perfect matching.



1 P e) Consider the following self-organizing list. Provide the number of key comparisons when the elements 'L', 'A', 'M', 'A', 'H', 'A', 'A', 'R' are accessed in this order using the Move-to-Front rule.

$$\mathrm{A} \longrightarrow \mathrm{L} \longrightarrow \mathrm{G} \longrightarrow \mathrm{O} \longrightarrow \mathrm{R} \longrightarrow \mathrm{I} \longrightarrow \mathrm{T} \longrightarrow \mathrm{H} \longrightarrow \mathrm{M} \longrightarrow \mathrm{U} \longrightarrow \mathrm{S}$$

1 P f) Insert the key 7 into the following 2-3 tree (B tree of order 3). After that, insert the key 20 into the resulting tree. Perform also the necessary structural changes.



After inserting 7:	After inserting 20:

1 P g) Specify an **order** for the functions below such that the following holds: If function f is left of function g, then $f \in \mathcal{O}(g)$.

Example: The three functions n^3, n^7, n^9 are already in a correct order, since $n^3 \in \mathcal{O}(n^7)$ and $n^7 \in \mathcal{O}(n^9)$.

- 2^n
- $\binom{n}{2}$
- $n \cdot (\log n)^5$
- 15^7
- $3^{n/2}$
- $n!$
- $\frac{n}{(\log n)^2}$

3 P h) Consider the following recursive formula:

$$T(n) := \begin{cases} 12 + 7T(n/7) & n > 1 \\ 3 & n = 1 \end{cases}$$

Specify a closed (i.e., non-recursive) form for $T(n)$ that is *as simple as possible*, and prove its correctness using mathematical induction.

Hints:

(1) You may assume that n is a power of 7.

(2) For $q \neq 1$, we have $\sum_{i=0}^{k} q^i = \frac{q^{k+1}-1}{q-1}$.

1 P i) Specify (as concisely as possible) the asymptotic running time of the following code fragment in Θ notation depending on $n \in \mathbb{N}$. You do not need to justify your answer.

```
1 for(int i = 1; i <= n; i += 3) {
2         for(int j = 1; j <= 2*i; j ++)
3                 ;
4 }
```

1 P j) Specify (as concisely as possible) the asymptotic running time of the following code fragment in Θ notation depending on $n \in \mathbb{N}$. You do not need to justify your answer.

```
1 for(int i = 1; i < n; i ++) {
2         for(int j = n; j >= 3; j /= 3)
3                 ;
4 }
```

1 P k) Prove or disprove: the minimum spanning tree of an undirected weighted graph $G = (V, E, w)$ is uniquely determined if and only if G does not contain two edges with the same weight.

3 P l) You are given an AVL tree of height h (reminder: an AVL tree of height 1 consists of exactly one node). We define $\phi := (1+\sqrt{5})/2$. Let $N(h)$ be the minimum number of nodes of an AVL tree of height h. Show by general induction over h that every AVL tree of height h has at least ϕ^{h-1} many nodes, i.e. show that $N(h) \geq \phi^{h-1}$ holds. For this purpose, your inductive step should contain a sketch describing the structure of an AVL tree of height h that has a minimum number of nodes.

Hint: We have $\phi^2 = 1 + \phi$.

Problem 2.

A student wants to cook fried potatoes, one portion today and one portion tomorrow. For this purpose he has n potatoes $\{1, \ldots, n\}$ available that have an overall weight of $G \in \mathbb{N}$ gram. The potato i weighs $g_i \in \mathbb{N}$ gram. We want to partition *all* n potatoes into the portions A and B such that these portions have roughly the same weight. Since the portions are cooked on different days, every potato must either be completely used for portion A, or completely used for portion B (every potato must either be used immediately, or must be *completely* saved for tomorrow). More concretely we want to compute two sets A and B of potatoes with $A \cap B = \emptyset$, $A \cup B = \{1, \ldots, n\}$ whose weight difference is minimum.

8 P a) Provide a dynamic programming algorithm for computing the minimum weight difference of two sets A and B (as described above). Address the following aspects in your solution.

1) What is the meaning of a table entry, and which size does the DP table have?
2) How can an entry be computed from the values of previously computed entries?
3) In which order can the entries be computed?
4) How can the value of the minimum weight difference be obtained from the DP table?

Hint: The trivial algorithm that simply enumerates all possible solutions does **not** give any points.

2 P b) Describe in detail how you can recognize from the DP table which potato is used on which day.

3 P c) Provide the running time of algorithm developed in a) and in b), and justify your answer. Is the running time polynomial?

Problem 3.

When trading with currencies, *arbitrage* means to exploit price differences for gaining money by changing currencies multiple times. For example, on June 2nd, 2009, 1 US Dollar could be changed to 95.729 Yen, 1 Yen to 0.00638 Pound sterling, and 1 Pound sterling in 1.65133 US Dollar. If a trader changed 1 US Dollar to Yen, the obtained amount to Pound sterling and finally changed this amount back to US Dollar, he would have obtained $95.729 \cdot 0.00638 \cdot 1.65133 \approx 1.0086$ US Dollar, corresponding to a gain of 0.86%.

3 P a) You are given n currencies $\{1, \ldots, n\}$ and an $(n \times n)$ exchange rate matrix $R \in (\mathbb{Q}^+)^2$. For two currencies $i, j \in \{1, \ldots, n\}$ one unit of currency i can be changed to $R(i, j) > 0$ units of currency j. The goal is to decide whether an arbitrage activity is possible, i.e., if there exists a sequence of k different currencies $W_1, \ldots, W_k \in \{1, \ldots, n\}$ such that $R(W_1, W_2) \cdot R(W_2, W_3) \cdots R(W_{k-1}, W_k) \cdot R(W_k, W_1) > 1$ holds.

Model the above problem as a graph problem. Show how the input can be transformed into a directed, weighted graph $G = (V, E, w)$ that contains a cycle with negative weight *if and only if* an arbitrage activity is possible. Justify why G contains a negative cycle if and only if an arbitrage activity is possible.

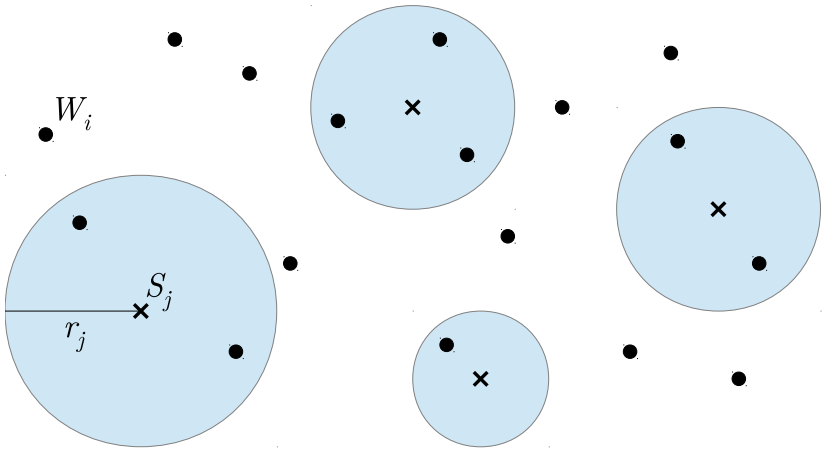
Notice: Using the logarithm might be beneficial because we have $\ln(a \cdot b) = \ln(a) + \ln(b)$.

3 P b) Which shortest path algorithm can be used for detecting cycles of negative weight? At which vertex can the algorithm start? Which running time (in dependency of n) does the chosen algorithm achieve if it is applied to the graph constructed in a)?

Problem 4.

You are given a two-dimensional map that shows an extract of some mountains and that contains the positions of n hikers. The hiker i is located at position $W_i = (x_i^W, y_i^W) \in \mathbb{Q}^2$. In the mountains there exist m mobile phone transmitters. The transmitter j has the position $S_j = (x_j^S, y_j^S) \in \mathbb{Q}^2$ and a range $r_j \in \mathbb{Q}^+$. Concretely this means that the mobile phones of all persons located inside a circle with origin S_j and radius r_j have service. Your task is to identify all hikers whose mobile phone has *no* service.

Example: In the map below the positions of the hikers are drawn as bold black points. The positions of the transmitters are the crossed centers of the circles, and the ranges correspond to the respective radiuses of the circles.



9 P a) Design a scanline algorithm that obtains as input the positions $W_1, \ldots, W_n$ of n hikers, the positions $S_1, \ldots, S_m$ of m transmitters and the respective ranges $r_1, \ldots, r_m$, and that computes the set of all hikers whose mobile phone has no service. Address the following aspects in your solution.

1) In which direction is the scanline moving, and what are the stopping points?
2) Which objects have to be stored in the data structure, and what is an appropriate choice for it?
3) What happens if the scanline encounters a new stopping point?
4) How can we finally identify all hikers whose mobile phone has no service?

Hint: For the sake of simplicity we assume that the ranges of no two transmitters overlap. As usual you can assume that no degenerate cases occur. This means, for example, that no two hikers have the same x or y coordinate.

1 P b) Provide the running time of the algorithm developed in a) and justify your answer.