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

January 25, 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 foreign language 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	5	**Σ**
max. score	9	8	8	12	13	50
$\sum$ 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) The following array contains the elements of a max-heap stored in the usual fashion. Specify the array after the maximum has been removed and the heap condition has been reestablished.

27	17	20	15	7	9	13	8	2	5	3	1	6
1	2	3	4	5	6	7	8	9	10	11	12	13

1 P (b) Let $G = (V, E)$ be a connected, undirected graph with $n = |V|$ vertices and $m = |E|$ edges. How many vertices and edges does a spanning tree of G have?

Vertices: ________ Edges: ________

1 P (c) Provide a connected graph with 7 vertices that has a maximum matching of size 2.

3 P (d) For each of the following statements, mark with a cross whether it is true or false. Every correct answer gives 0.5 points, for every wrong answer 0.5 points are removed. A missing answer gives 0 points. In overall the exercise gives at least 0 points. You don't have to justify your answer.

A postorder traversal of a binary search tree creates a descending ordered list of the stored keys.	☐ TRUE	☐ FALSE
There exists an AVL tree in which more than half of its inner vertices are not balanced (i.e., they have a balance factor different from 0).	☐ TRUE	☐ FALSE
For every AVL tree, there exists an insertion order leading to exactly this tree such that no rotations occur.	☐ TRUE	☐ FALSE
Even in the worst case, inserting into an AVL tree requires at most one (single or double) rotation.	☐ TRUE	☐ FALSE
Inserting a new element into a Fibonacci Heap requires only constant time in the worst case.	☐ TRUE	☐ FALSE
Insertion sort can be implemented as a stable sorting algorithm.	☐ TRUE	☐ FALSE

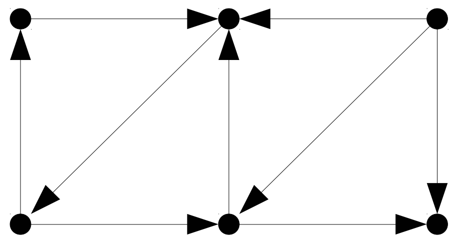
1 P (e) Consider a segment tree for integer intervals having integer boundaries from $\{1, \ldots, n+1\}$ where n is a power of 2 (i.e., $n = 2^k$, $k \in \mathbb{N}$). Provide the number of vertices that are visited at most in a query for a point $i \in \{1, \ldots, n+1\}$.

Maximum number: ________

1 P (f) Insert the keys 12, 19, 6, 15, 13, 2, 28 in this order into the following hash table. Use Double Hashing with the hash function $h(k) = k \bmod 11$ and resolve collisions using the function $h'(k) = 1 + (k \bmod 9)$.

0	1	2	3	4	5	6	7	8	9	10

1 P (g) In the following graph $G = (V, E)$, mark a smallest possible set S of edges such that $G' = (V, E \backslash S)$ can be sorted topologically.



Problem 2.

1 P (a) 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 $n^7 \in \mathcal{O}(n^9)$.

- $\frac{3^n}{n^3}$
- 10^{10}
- $\log(n^n)$
- $n!$
- $n^3 + n$
- $\sqrt{3^n}$

4 P (b) Consider the following recursive formula:

$$T(n) := \begin{cases} 5T(n/5) + n + 4 & n > 1 \\ 1 & n = 1 \end{cases}$$

Specify a closed form (i.e., non-recursive) 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 5.

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

1 P (c) 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.

```
for(int i = 1; i <= n; i += 10) {
        for(int j = 1; j <= n/2; j += 4)
                ;
}
```

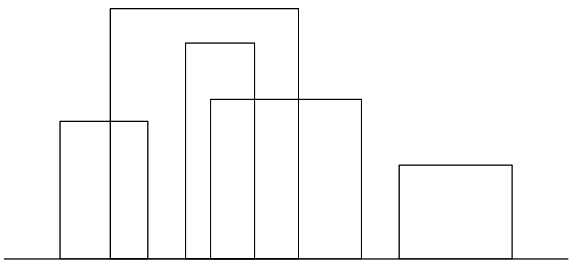
1 P (d) Specify (as concisely as possible) the asymptotic running time of the following code fragment in Θ notation depending on $n \in \mathbb{N}$. You do not need to justify your answer.

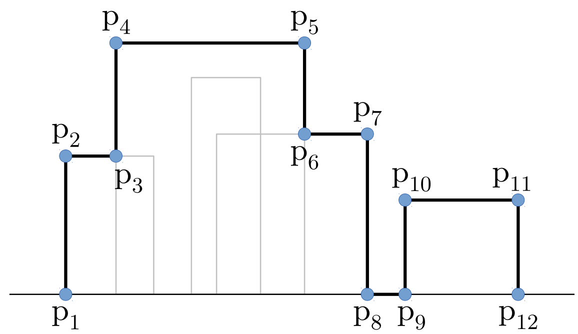
```
for(int i = 1; i <= n; i++) {
        for(int j = 1; j*j <= n; j++)
                ;
        for(int k = n; k ≥ 2; k /= 2)
                ;
}
```

1 P (e) 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, and you can assume that n is a power of 2.

```
int f(int n) {
        if(n <= 1) { return 1; }
        else { return f(n/2)+f(n/2); }
}
```

Problem 3. This exercise is concerned with the computation of contours of skylines. A skyline of a city (the contour) is the shadow of a set of rectangular high-rise buildings that sit on the ground and that are provided as a set of orthogonal rectangles as shown in the image on the left. The contour is the orthogonal polygon describing the union of the given rectangles as shown in the image on the right. The contour can be described by the set of its corners $p_1, \ldots, p_k$.





7 P a) Provide an efficient sweepline algorithm that obtains a set of n orthogonal rectangles as input, and that computes the corners $p_1, \ldots, p_k$ of the contour. Address the following aspects in your solution.

1) In which direction is the sweepline 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 sweepline encounters a new stopping point?

4) How can the solution be extracted?

1 P b) Provide the running time of your algorithm and justify it.

Problem 4. An online trading offers the article types $\{1, \ldots, n\}$. We order some examples of every article type, and we commission $m \leq n$ couriers to deliver them. However, not every courier can transport every article type (i.e., a bicycle courier cannot deliver a TV set). For every courier $j \in \{1, \ldots, m\}$, let $T(j) \subseteq \{1, \ldots, n\}$ be the set of all article types that can be delivered by j. We want to decide whether it is possible to assign the article (examples) to the couriers such that all articles can be delivered, and every courier has to travel only once.

4 P a) Suppose that we order exactly one example of each each article type $i \in \{1, \ldots, n\}$, and that every courier $j \in \{1, \ldots, m\}$ can only deliver one example per trip. Model the above problem as a flow problem. Describe the construction of an appropriate network $G = (V, E, c)$ with the vertex set V and the edge set E, and which capacities have to be assigned to the edges. How can you deduce from the value of a maximum flow whether an appropriate assignment exists, or not?

3 P b) Now we order exactly $f(i) \in \mathbb{N}_0$ examples of article type $i \in \{1, \ldots, n\}$ (instead of just one), and every courier $j \in \{1, \ldots, m\}$ can deliver up to $k(j)$ article examples per trip (instead of one). Describe how your solution from a) can be modified to solve this generalized problem. How large does the value of a maximum flow have to be such that every courier has to travel only once?

Example:

i	Article type	Ordered examples $f(i)$
1	Table	2
2	Chair	2
3	Pen	5
4	Printer	0

Courier j	Max. no. of articles $k(j)$ per travel	Accepted article types $T(l)$
1	4	$\{3\}$ (Pen)
2	3	$\{1, 2, 3, 4\}$ (Everything)
3	2	$\{1, 2\}$ (Table, Chair)

Here, one trip per courier is sufficient: 4 pens can be delivered by courier 1, 1 pen and 2 tables can be delivered by courier 2, and 2 chairs can be delivered by courier 3.

2 P c) Specify a flow algorithm that solves b) as efficient as possible. Provide the running time of the above-mentioned algorithm in dependency of the number of article types n and the number of couriers m. Justify your answer.

3 P d) Suppose that we already solved the flow problem described in b), i.e., we know the value of flow ϕ_e for each edge e, and an assignment of articles and couriers really exists. Describe in detail an algorithm that uses the ϕ_e values and computes such an assignment. Concretely, we want to compute a set M that contains a triple (j, i, k) iff the courier j delivers exactly k examples of article i. For the above example we have $M = \{(1, 3, 4), (2, 1, 2), (2, 3, 1), (3, 2, 2)\}$. What is the running time of your algorithm if every ϕ_e can be accessed in constant time?

Problem 5. In Switzerland, you have the following coin values: 5 Rappen, 10 Rappen, 20 Rappen, 50 Rappen, 1 Franken, 2 Franken, 5 Franken. Using these coins, you can represent all amounts of money whose value in Rappen can be divided by 5. This exercise is concerned with the number of different possibilities to achieve a given amount by some set of coins. For example, the amount of 20 Rappen can be achieved on four different ways:

1) $5 + 5 + 5 + 5$ Rappen,
2) $5 + 5 + 10$ Rappen,
3) $10 + 10$ Rappen,
4) 20 Rappen.

Notice that the order of the coins is not relevant, i.e. $5 + 5 + 10$ Rappen is the same set of coins than $5 + 10 + 5$ Rappen. For the exercise we assume that we are given n coins with the values $M_1, \dots, M_n \in \mathbb{N}$ (in the same unit, e.g., Rappen). W.l.o.g., let them be pairwise different and ordered increasingly, i.e. $M_1 < M_2 < \cdots < M_n$. For example, for the Swiss currency, we have the coin values $M_1 = 5$, $M_2 = 10$, $M_3 = 20$, $M_4 = 50$, $M_5 = 100$, $M_6 = 200$, $M_7 = 500$ (in Rappen).

Notice: The coin values M_i are not necessarily multiples of 5 like in the example above, but arbitrary numbers. It might even be that all M_i are mutually prime.

8 P a) Let $B \in \mathbb{N}$ be an amount of money in the same unit as the coins $M_1, \dots, M_n$. Provide an efficient algorithm that uses dynamic programming to compute the number of different possibilities to represent B with the coins $M_1, \dots, M_n$. 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 other entries?
3) In which order do the entries have to be computed?
4) How can the final solution be extracted once the table has been filled?

2 P b) Provide the running time of your solution of a) and justify it. Is it polynomial?

3 P c) Describe in detail how *one* possible representation of B (if exists) can be found by using the solution table. Provide the running time of your algorithm.