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 24, 2013

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 record 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 120 minutes to solve the exam.

Good luck!

Student number: _____

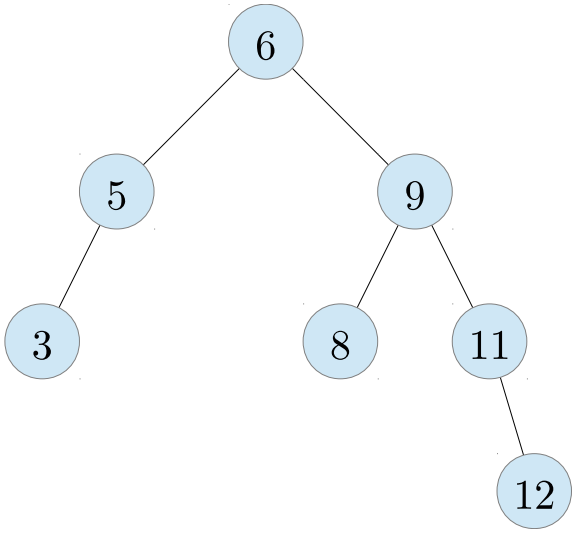
problem	1	2	3	4	5	**Σ**
max. score	8	7	9	9	10	43
$\sum$ score						

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) Insert the key 15 in the AVL tree below. *After that*, remove the key 3 from the resulting tree.



After insertion of 15:	After deletion of 3:

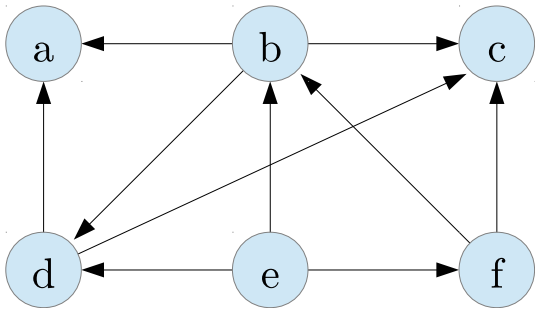
1 P (b) The following array contains the elements of a max-heap stored in the usual fashion. Specify the array after the maximum has been removed and the heap condition has been reestablished.

35	33	22	20	24	10	16	12	11	17	14	9	5
1	2	3	4	5	6	7	8	9	10	11	12	13

1 P (c) Insert the keys 16, 6, 24, 41, 18, 38, 28 in this order into the following hash table. Use open hashing with the hash function $h(k) = k \bmod 11$ and resolve collisions by quadratic probing.

0	1	2	3	4	5	6	7	8	9	10

1 P (d) Provide a topological sorting of the graph below.



Topological sorting: _____, _____, _____, _____, _____, _____

1 P (e) Provide a connected graph with 6 vertices that contains **exactly** 3 different perfect matchings.

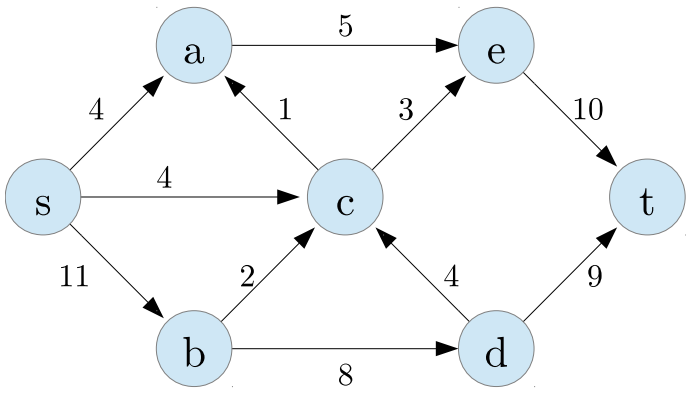
1 P (f) How many edges can a bipartite graph with n vertices have at most? For the sake of simplicity, you can assume that n is even.

1 P (g) Provide the key pairs that are considered in the first four comparisons when the following array is sorted with *insertion sort*.

8	15	6	12	4	9	10	11	13	19
1	2	3	4	5	6	7	8	9	10

Key pairs: (____, ____), (____, ____), (____, ____), (____, ____)

1 P (h) Consider the following network with source s and target t. The individual capacities are given by the numbers next to the edges.



A flow ϕ is partially given by the following function:

(v, w)	(s, a)	(s, c)	(b, d)	(c, e)	(d, t)
$\phi(v, w)$	4	4	8	3	8

Complete the values on the remaining edges such that ϕ is a valid flow. In any case, this flow will be maximal. Draw a cut in the image above that shows the maximality of ϕ.

(v, w)	(s, b)	(a, e)	(b, c)	(c, a)	(d, c)	(e, t)
$\phi(v, w)$						

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

- n^2
- $\binom{n}{4}$
- $\log(n^{17})$
- $(\log n)^5$
- $n!$
- $\sqrt{3^n}$
- 10^{50}

3 P (b) Consider the following recursive formula:

$$T(n) := \begin{cases} 8 + 5T(n/3) & n > 1 \\ 3 & n = 1 \end{cases}$$

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 3.

(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.

```
1 for(int i = 0; i <= n; i ++) {
2         for(int j = 1; j <= n*n*n; j *= 2)
3                 ;
4         for(int k = 1; k <= n; k += 2)
5                 ;
6 }
```

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.

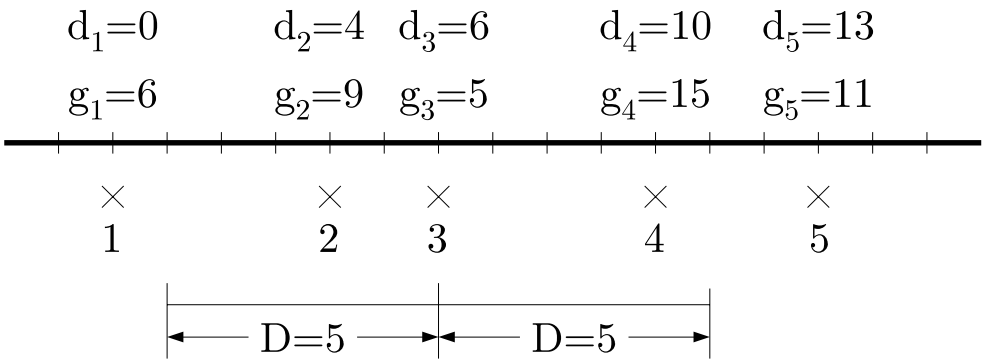
```
1 for(int i = 1; i <= ⌈log_5(n)⌉; i ++) {
2         int k = n;
3         while(k > 1)
4                 k = k/4;
5 }
```

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.

```
1 int f(int n) {
2         if(n == 1) return 1;
3         else {
4                 for(int i = 1; i <= n; i ++)
5                         ;
6                 return f(n/3)+1;
7         }
8 }
```

Problem 3.

We want to place wind turbines along a road to produce energy. Due to geographical reasons n different positions are possible, but laws prescribe that the distance between two wind turbines has to be at least D. The possible positions $d_1, ..., d_n$ are given as coordinates on a line, where the leftmost position has the value 0. In other words: The distance between the i-th possible position and the first possible position is d_i, we have $d_1 = 0$, and $d_i < d_{i+1}$ for each $i \in \{1, ..., n-1\}$. When a wind turbine is installed at position i, the profit is $g_i > 0$. The task is to find a positioning of wind turbines that maximizes the profit.



Example: The image above shows a situation for $n = 5$ possible positions. For example, if a wind turbine is installed at position 3, then no wind turbines can be installed at the positions 2 or 4. When the wind turbines are placed on the positions 1, 3 and 5, then the profit is $6 + 5 + 11 = 22$. This solution is not optimal: An installation of wind turbines on the positions 2 and 4 has a profit of $9 + 15 = 24$.

1 P (a) Provide an example (as simple as possible) that shows that the following greedy strategy does not necessarily lead to an optimal solution: *"Select a possible position with maximal profit until no other wind turbines can be placed."*

5 P (b) Describe a *dynamic programming* algorithm that computes the maximal profit as efficiently as possible.

1 P (c) Specify the running time of your solution.

2 P (d) Describe in detail how the above algorithm has to be modified to compute an optimal placement of wind turbines with a maximal profit.

Problem 4.

Consider a university with n students. Every student has to attend exactly 5 courses in the next semester. In overall there exist m courses, and the course k can be attended by at most $T_k \in \mathbb{N}$ students.

Every student i is interested in a set K_i of 10 courses. The university wants to assign courses to the students such that every student attends only courses he likes, and that no course k is attended by more than T_k students.

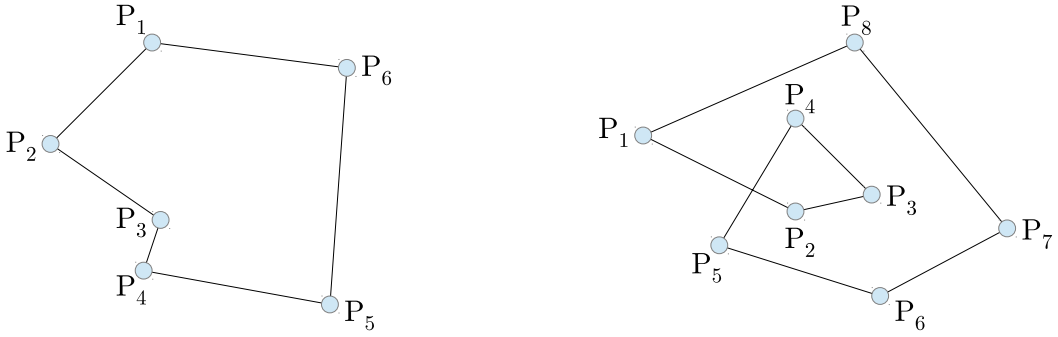
4 P (a) First, we want to decide whether such an assignment of courses to students exists. Model this problem as a flow problem. Explain carefully how a network $N = (V, E)$ can be constructed from $K_1, ..., K_n$ and $T_1, ..., T_m$. Specify the vertices V and the edges E of the construction and describe which capacities have to be assigned to the edges. Describe how you can conclude from the value of a maximal flow whether such an assignment exists or not.

2 P (b) Specify a flow algorithm that solves (a) as efficient as possible. Assume that there are more students than courses, and specify the running time of the above-mentioned algorithm in dependency of n and m.

3 P (c) Assuming that an appropriate assignment of students to courses exists, we are interested in computing this assignment. Describe an algorithm that computes for every student i the set of the 5 courses that i has to attend in the next semester. Which running time does your method have, if a maximal flow in the network N has been already computed before?

Problem 5.

Let $P_1, ..., P_n$ be a set of points in $\mathbb{R}^2$. When a line segment connects the point P_i to P_{i+1} for every $i \in \{1, ..., n-1\}$ and additionally P_n to P_1, we obtain a *polygon*.



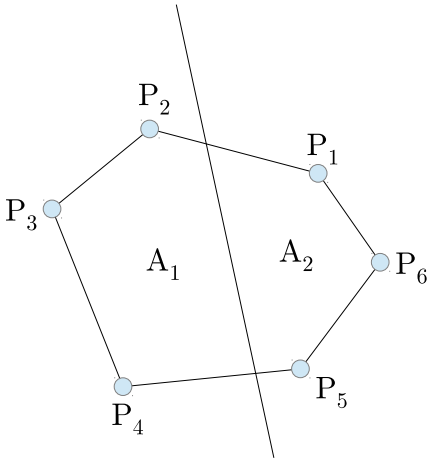
4 P (a) A polygon P is called *simple*, if no two edges of P cross. In the image above, only the polygon on the left is simple. The polygon on the right is not simple, because the segments $\overline{P_1P_2}$ and $\overline{P_4P_5}$ cross.

Provide a scanline algorithm that determines in time $\mathcal{O}(n \log n)$ whether a polygon is simple.

4 P (b) We will now assume that $P_1, ..., P_n$ forms a convex polygon (Reminder: A polygon P is called *convex* if for any two points in P, the line segment connecting these points is contained entirely in P).

Provide an algorithm that computes the area of the polygon defined by $P_1, ..., P_n$ as efficiently as possible. Which running time does your solution have?

2 P (c) Like in (b) we assume that the polygon defined by $P_1, ..., P_n$ is convex. Additionally, we are given the parameters m and b of a line $y = mx + b$ that splits the polygon into two parts A_1 and A_2.



Describe an efficient algorithm that uses the results from task (b) to compute the area of the parts A_1 and A_2. Specify the running time of your solution.