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
Holger Flier

Prüfung
Datenstrukturen und Algorithmen
D-INFK

January 26, 2012
English version

Family name, first name: ______________________

Student-Number: ______________________

With my signature, I confirm that I could take this exam under regular circumstances and that I have read and understood the remarks below.

Signature: ______________________

Remarks:

- Except for a dictionary, no auxiliary means are allowed.
- Please write your student number on **each** sheet.
- Please notify us **immediately**, if there is any disturbance which has an adverse effect on your work.
- Please use a separate sheet of paper for each task. Only one solution per task can be handed in. Invalid tentative solutions have to be crossed out clearly.
- Please write **legibly** with blue or black ink. We only give points for legibly written text.
- The exam takes 120 minutes. **Don't worry!** We don't expect that anyone can solve all the tasks in the given time! The maximum grade can be achieved with far less than the total of achievable points.

Good luck!

Student-Number: ______________________________

Task	1	2	3	4	5	$\boldsymbol{\Sigma}$
Possible Points	9	7	9	9	9	43
Points						

Task 1:

Remarks:

1. In this task it suffices to write down **only the results**. These can be written directly below the task descriptions.
2. Provided that you use notations, algorithms and data structures from the lecture "Datenstrukturen & Algorithmen", no explanations or argumentation is required. If you use other methods, you should **briefly** explain your results to make them comprehensible and traceable.
3. We order letters by their alphabetic order, and numbers in increasing value.

1 P a) The algorithm of Karatsuba/Ofman for the multiplication of integers computes the product of two numbers recursively according to a formula which contains, aside from additions and multiplications with the basis (here: 10), three products. Give two numbers x and y, for which these three products are $(12 \cdot 34)$, $(56 \cdot 78)$ and $(12 \pm 78) \cdot (56 \pm 34)$.

$x =$ ____________________, $y =$ ____________________

1 P b) Execute the pivot step of the sorting algorithm Quicksort on the given array (in-situ, i.e., without an auxiliary array). Use the rightmost element of the array as the pivot element.

25	12	83	2	58	68	19	34	47	99	37	56	41

1 P c) The array below contains the elements of a min-heap in the usual order. Where are the elements in the array after the minimum has been removed and the heap-property has been restored?

4	24	10	38	30	72	77	75	78
1	2	3	4	5	6	7	8	9

1 P d) Assume that the keys $1, 7, 42, 46, 35, 41, 18, 9, 30, 28, 31, 24, 19, 15, 27$ have to be inserted in this order into hash tables t_1 and t_2 using cuckoo hashing. The hash function for table t_1 is $h_1(k) = k \mod 7$, the one for table t_2 is $h_2(k) = 3k \mod 7$. Assume that t_1 and t_2 both have length 7. What is the first key in the sequence above that cannot be inserted without enlarging the tables (i.e., performing a `rehash` operation)?

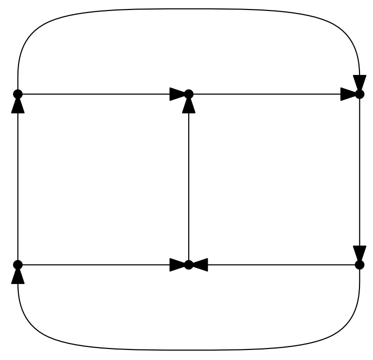
Key: _____

1 P e) Draw a tree that has an odd number of edges and a minimum number of nodes such that it has a maximum independent set of cardinality 4.

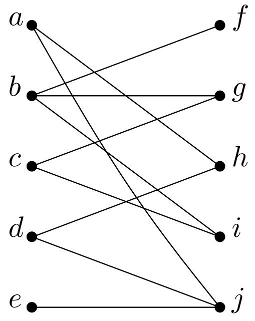
1 P f) Below there are eight keys together with their frequencies. Create a coding tree using the algorithm of Huffman and draw the resulting tree.

Schlüssel	a	b	c	d	e	f	g	h
Anzahl Zugriffe	31	23	60	29	13	15	68	88

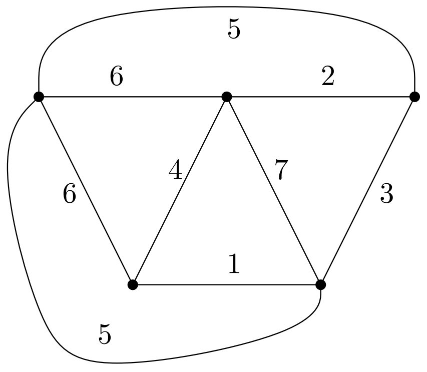
1 P g) In the following graph $G = (V, E)$, indicate a smallest possible set S of edges, such that $G' := (V, E \setminus S)$ has a topological sort (topological ordering).



1 P h) Give a subset of the vertices of the following bipartite graph which proves, together with Hall's theorem, that the graph does not contain a perfect matching.



1 P i) Which edge of the following graph is the first to be rejected during the computation of a minimum spanning tree (MST) when using Kruskal's algorithm?



Task 2:

1 P a) Provide an **order** of the functions given below, which has the following property: If the function f stands left of the function g then it holds that $f \in O(g)$.

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

- $30\sqrt{n}$
- $3^{\frac{n}{2}}$
- $n!$
- $10n$
- $20\log(n)$
- $n\log n$
- $n^{\frac{2}{3}}$

3 P b) Consider the following recursive equation:

$$T(n) := \begin{cases} 4 + 2T(\frac{n}{8}) & n > 1 \\ 1 & n = 1 \end{cases}$$

Give a closed (i.e., non-recursive) expression for $T(n)$ (simplified as far as possible) and prove its correctness using induction.

Hints:
(1) You may assume that n is a power of 8.
(2) For $q \neq 1$ it holds: $\sum_{i=0}^{k} q^i = \frac{q^{k+1}-1}{q-1}$.

1 P c) Determine the asymptotic running time of the following code, depending on $n \in \mathbb{N}$, expressed in Theta-notation (you don't need to justify your answer):

```
from j := 1 until j > n loop
   from k := 2 until k > n loop
      k := k * 2
      from l := 1 until l > n loop
         l := l + 10
      end
   end
   j := j + 1
end
```

1 P d) Determine the asymptotic running time of the following code, depending on $n \in \mathbb{N}$, expressed in Theta-notation (you don't need to justify your answer):

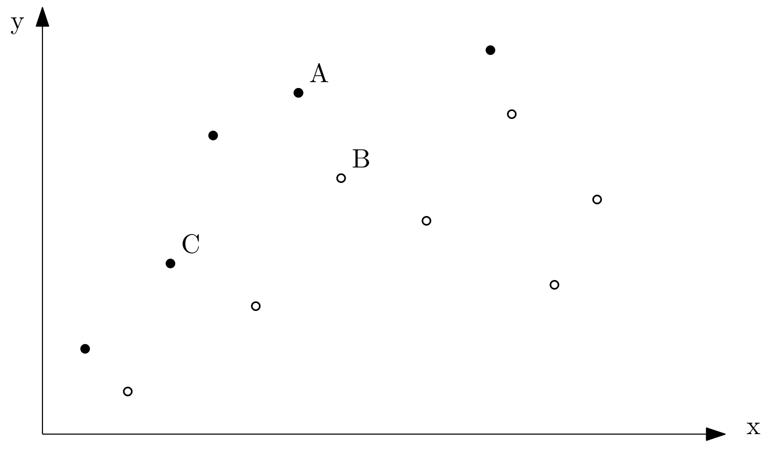
```
from h := 1 until h > n loop
   from j := 1 until j > n*n loop
      j := j * 3
   end
   from k := 2 until k*k > n loop
      k := k + 1
   end
   h := h + 2
end
```

1 P e) Determine the asymptotic running time of the following code, depending on $n \in \mathbb{N}$, expressed in Theta-notation (you don't need to justify your answer):

```
from j := 1 until j > n loop
   from k := 1 until k > n*n*n loop
      k := k + j
   end
   j := j + 2
end
```

Task 3:

An online shop for smartphones seeks to offer its customers only the best devices. Two important criteria are the price and the maximal operating time (of the battery) of the smartphone. The shop maintains a database in which these information are stored for every smartphone. Only the dominant smartphones should be suggested to the clients: if, e.g., a device A for the price of 300CHF offers an operating time of 8h and another device B for the price of 350CHF offers only an operating time of 6h, then B is not dominant.
Formally, we consider a set of points in the plane, where each point (x, y) represents a smartphone, and where x denotes the price and y the operating time. We assume that the points lie in general position, i.e., no two points have the same $x-$ or $y-$coordinate. We say that a point (x, y) *dominates* another point (x', y'), if both $x < x'$ and $y > y'$ hold. A point is called *dominant*, if it is not dominated by any other point.
In the following figure point A dominates point B. Point C dominates neither A nor B. The dominating points are depicted in black.



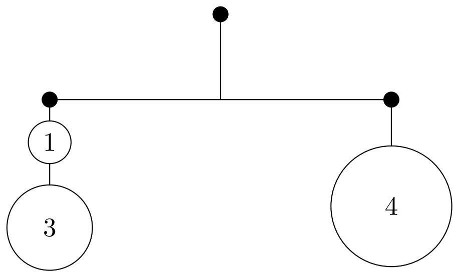
3 P a) Design an algorithm, as efficient as possible, that computes a list of dominating points for a given set of n points. Describe your algorithm in words and/or pseudocode. Specify the running time of your algorithm in the worst case.

6 P b) Now we consider a dynamic version of the problem solved in a), in which points may be added. No points may be removed, however. Hence, you should design a data structure that supports the following operations as efficiently as possible (with regard to the number of points in the current set):

- `Init()`: creates a data structure that represents an empty set of points
- `Insert`(x, y): adds a point (x, y) to the current set of points
- `ListDominant()`: returns the set of dominant points in the current set of points

Describe your data structure in words and the implementation of the three operations above in words and/or pseudocode. Specify the running time of each operation.

Task 4:
Alexander (an artist) is planning a mobile on the topic "balance" (see the figure below). He has n discs at his disposal, which can be hung below each other, forming chains of arbitrary length. The discs can have different weights: Each disc i has an integer weight w_i. The available n discs are to be distributed on the two sides (A and B) of the mobile, such that an equilibrium results. This is achieved exactly if the sum of the weights of the discs in A is equal to the sum of the weights of the discs in B. Note that *all* discs must be used.



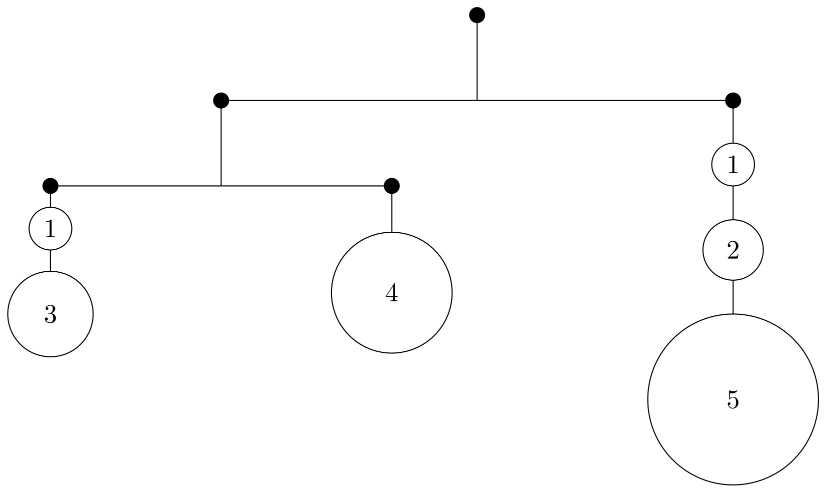
5 P a) Design an algorithm based on dynamic programming, which decides whether an equilibrium is possible for a given set of n discs with weights $w_1, w_2, \ldots, w_n$. Write down the recursive equation of this dynamic program, and state the running time of your algorithm.

1 P b) Briefly describe in words how the algorithm in a) can be extended, such that it outputs an actual partition of the discs achieving an equilibrium, provided that one exists.

3 P c) Since the first sculpture was a success, Alexander plans another one (see the figure below). For this one, the discs have to be split into three sets A, B and C, such that the following conditions hold:

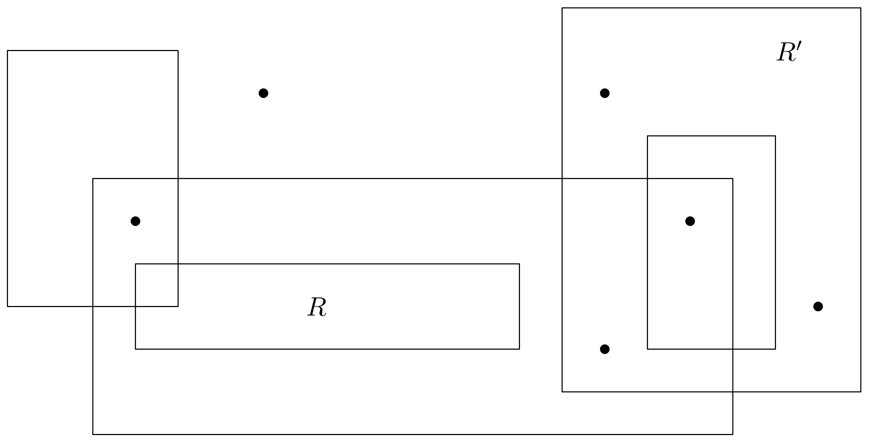
- The total weight of set A is equal to the total weight of set B.
- The total weight of set C is equal to the total weight of set A plus the total weight of set B.

Again, all discs must be used. Design a dynamic program for deciding whether such a partition is possible, and state the running time of your algorithm.



Task 5:

We are given a set of points in the plane as well as a set of iso-oriented (axis-parallel) rectangles. We want to determine which of the rectangles do not cover any of the points. In the figure shown below, this is only the case for rectangle R.



In the following, assume that a set of m iso-oriented rectangles and a set of n points are given as input. In particular, each point P_j, $j \in \{1, \ldots, n\}$, is described by coordinates (x_j, y_j) and each rectangle R_i, $i \in \{1, \ldots, m\}$, by its top-left corner (l_i, t_i) and its bottom right corner (r_i, b_i). You may assume that all coordinates are integer and are within the range $\{1, \ldots, m + n\}$.

5 P a) Design an algorithm, as efficient as possible, which reports all rectangles that do not cover any of the points. Describe your algorithm in words and/or pseudocode.

2 P b) Which running time does your algorithm have in the worst case, depending on m, n and the number of reported rectangles?

2 P c) Next, we want to identify a rectangle which covers the maximum number of points among all rectangles. Note that there may be several such rectangles. Note further that a point may lie within several rectangles at once. In the example above, the rectangle we are looking for is R' since no other rectangle contains more points. Design an algorithm, as efficient as possible, which outputs such a rectangle, and describe your algorithm in words and/or pseudocode.