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
Beat Gfeller

Prüfung
Datenstrukturen und Algorithmen
D-INFK

19. August 2009

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 **legible** 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: _____

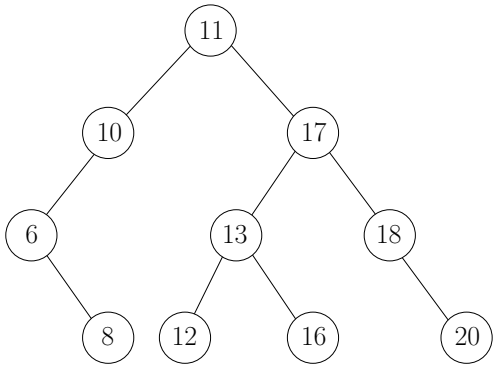
Task	1	2	3	4	5	**Σ**
Possible Points	9	7	9	9	9	43
Points						

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) Write down the preorder sequence of the nodes in the following tree:



1 P b) The array below contains the elements of a Min-heap, stored in the usual way. How does the array look like after the minimum has been deleted and the heap order has been restored?

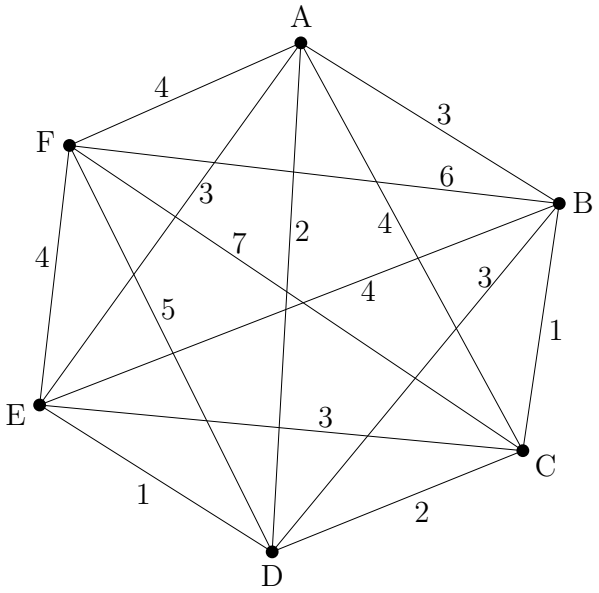
1	5	7	11	9	10	8	12	18	42

1	2	3	4	5	6	7	8	9	10

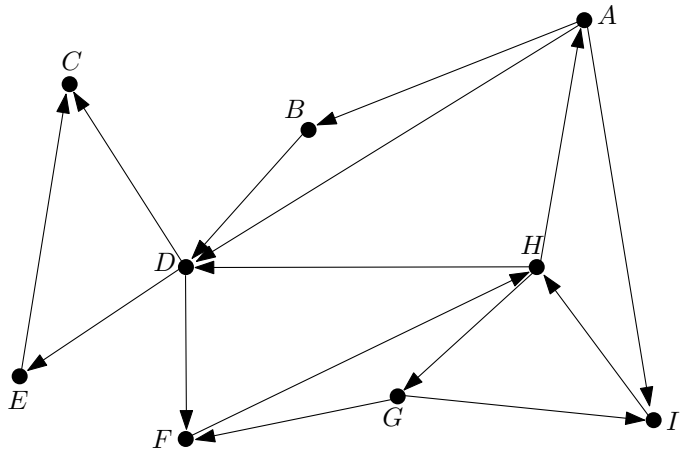
1 P c) Execute one partitioning step (in situ, i.e., without an auxiliary array) of the sorting algorithm Quicksort on the following array. Use the rightmost element in the array as the pivot.

42	81	16	9	5	4	19	23	80	13	69	8	33

1 P d) In the following graph, use *Christofides' Algorithm* to compute a tour whose length is at most $\frac{3}{2}$ times as long as the optimal tour. You need not prove this bound. Write down the order in which the nodes are visited in the computed tour.



1 P e) The following directed graph is traversed using depth-first-search. The search starts at node *A*. Write down a possible sequence in which the nodes are visited.

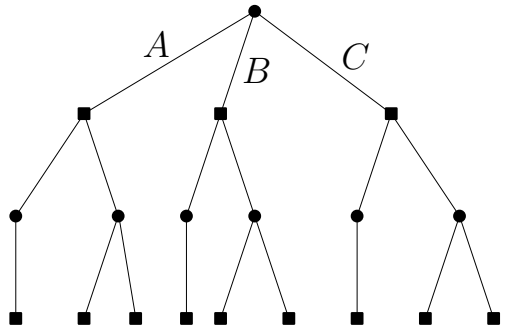


1 P f) Below, we give eight keys with their corresponding number of accesses. Using the Huffman-algorithm, construct an optimal code tree, and draw it.

Key	a	b	c	d	e	f	g	h
Number of accesses	22	1	5	3	3	2	1	9

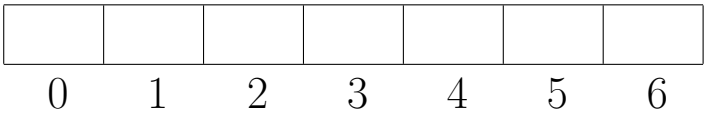
1 P

g) The following tree represents a game tree search of depth three, where a value is assigned to every possible resulting state. A higher value means that the state is better for the current player (and worse for the second player). Which next move (A, B or C) is best for the current player?



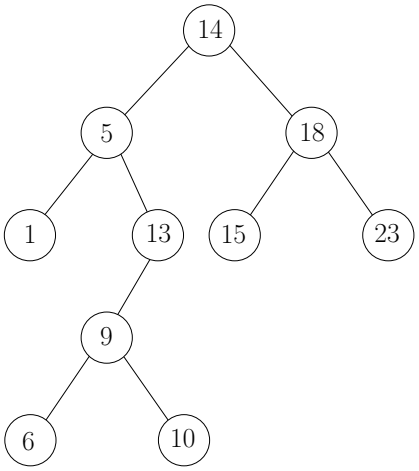
1 P

h) Enter the keys $1, 12, 4, 17, 89, 11, 5$ in this order into the following hash table with open hashing, and use quadratic hashing. The hash function to be used is $h(k) = (k \mod 7)$.



1 P

i) Draw the Splay-tree which results from the one shown below, after the node with key 9 has been accessed (and the corresponding splay-operations have been performed).



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 $n^7 \in O(n^9)$.

- $\log\left(\frac{n^3}{2}\right)$
- $2^{\sqrt{n}}$
- $\sum_{i=1}^{n} i$
- $\log^2 n$
- $\sqrt{n}$
- $\frac{n}{\log n}$

3 P b) Consider the following recursive equation:

$$T(n) := \begin{cases} 2T(\frac{n}{4}) + 3n + 5 & n > 1 \\ 2 & n = 1 \end{cases}$$

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 4.
(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 i := n until i < 1 loop
  from j := 1 until j > i loop
    x := i + 1
    j := j + 1
  end
  i := i - 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
  i := n
  s := 1
  k := 1
until i < 1 or i > 2*n loop
  i := i + s*k
  s := -s
  k := 2*k
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
  i := 1
  k := 1
until i > n loop
  from j := 1 until j > k loop
    j := j + 1
  end
  if k > n then
    k := k // 2
  else
    k := k * 2
  end
  i := i + 1
end
```

Task 3:

A hotel requires a system for handling the bookings of its president suite. This suite can be rented for an arbitrary number of nights. Of course, for every night the suite can only be rented by at most one client. The suite can be booked seamlessly, i.e., if a client checks out on day y, another client can check in on the same day. The system must support three operations: Entering bookings (when a reservation is made), deleting a booking (when a reservation is cancelled), and checking whether the suite is still available for a given pair of arrival day and departure day. A *booking* (x, y) consists of an arrival day x and a departure day y, where x and y are natural numbers, and $x < y$.

3 P a) As a first planning step, provide a data structure for storing n bookings that have already been made, in such a way that the question whether a desired booking (x', y') is still possible can be answered efficiently. Briefly describe in words

- how your data structure is composed,
- how it is constructed for n existing bookings, and how much time this construction requires,
- and how a query for an arbitrary booking (x', y') can be answered efficiently. Also state the running time of such an arbitrary query.

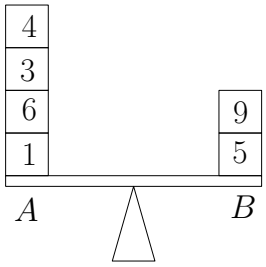
6 P b) We now want to generalize the system, such that the set of bookings is no longer static. This means that the data structure must provide the following operations on the (initally empty) set of bookings, in an efficient way:

- `Insert`(x, y): insert a booking (x, y)
- `Delete`(x, y): delete a booking (x, y)
- `Query`(x, y): decide whether a booking (x, y) is still possible given the current set of bookings

All operations should be as efficient as possible in the number of bookings in the current set. Describe your data structure in words, and the implementation of all three operations in pseudocode. Your pseudocode must be written in notation similar to Java, Eiffel or C++. Furhermore, state the running times of the operations `Insert`, `Delete` and `Query`.

Task 4:

An artist is planning a sculpture on the topic "balance" (see the figure below). She has n cubes at her disposal, which can be piled up on top of each other arbitrarily high. All cubes have the same height, but different weights: Each cube i has an integer weight w_i. The available n cubes are to be distributed on the two sides (A and B) of a scale, such that an equilibrium results. This is achieved exactly if the sum of the weights on the left side is equal to the sum of the weights on the right side. Note that *all* cubes must be used.



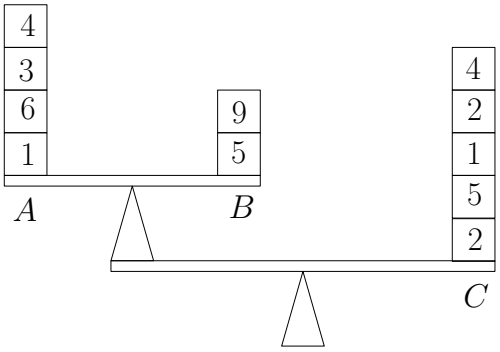
5 P a) Design an algorithm based on dynamic programming, which decides whether an equilibrium is possible for a given set of n cubes with weights $w_1, w_2, \ldots, w_n$. Write down the recursive equation of this dynamic program, and state the running time of your algorithm.

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

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

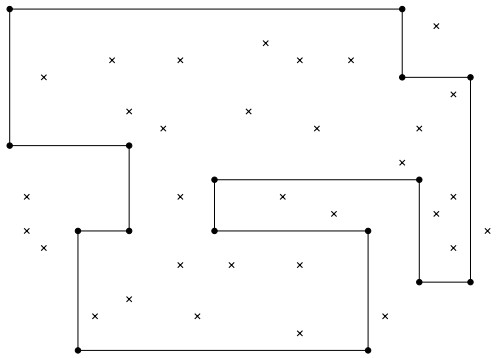
- The total weight of set A is equal to the total weight of set B.
- The total weight of set C is equal to the total weight of set A plus the total weight of set B.

Again, all cubes 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:

A Swiss farmer owns a large number of cows. These are all kept inside a single enclosure with straight fences. In order to check whether all cows are still inside the enclosure, the farmer wants to count the cows inside the enclosure. We assume that the farmer can determine the positions of all his cows, e.g. with the aid of a GPS-receiver for each cow. Of course, the farmer also knows the exact positions of the corners of his fence. For a start, we assume that every corner of the fence forms a right angle, and hence all edges are orthogonal to each other.
In this task, an algorithm for counting the (current) number of cows inside the enclosure shall be developed. More precisely: The input is a simple, closed polygon with n *orthogonal* edges (the enclosure), as well as the position of m points in the plane (the cows). The cows are described by their x and y coordinate. The polygon is given by the (unsorted) list of all edges, where each edge is given as an ordered pair of corners (a, b), where the inside of the polygon lies to the right, if one looks from a towards b. The output is the number of points that lie inside the polygon. For simplicity, we assume that no point lies exactly on the border of the polygon. An example instance is displayed on the figure below.



5 P a) Design an algorithm which, given an orthogonal polygon with n edges and a set of m points, computes the number of points inside the polygon, as efficiently as possible. Describe your algorithm in words and/or in pseudocode.

1 P b) What running time does your solution have in the worst case, in terms of m and n?

3 P c) Now the algorithm shall be extended, such that the edges of the polygon need not be orthogonal anymore. Describe in words which parts of your solution need to be extended, and how. What running time does the extended solution have?