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
Markus Püschel
Thomas Tschager
Tobias Pröger

Exam
Algorithmen und Datenstrukturen
D-INFK

December 15, 2016

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 120 minutes to solve the exam.

Good luck!

Student number: ______________________________

problem	1	2	3	$\mathbf{\Sigma}$
max. score	14	6	13	33
$\sum$ score				

/ 14 P

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 keys 4, 16, 20, 6, 12, 9, 5 in this order into the hash table below. Use open hashing with the hash function $h(k) = k \bmod 11$. Resolve collisions using quadratic probing. In case of a collision, first try probing to the left and only after that, to the right.

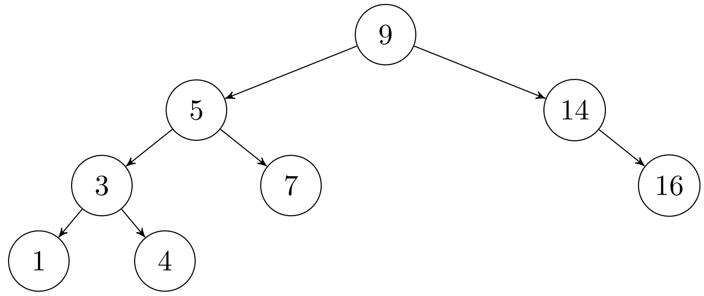
0	1	2	3	4	5	6	7	8	9	10

/ 1 P

b) Let $\mathcal{K} = \{5, 9, 8, 11, 15, 7, 20\}$ be a set of keys. Draw the two binary search trees that manage exactly the keys in $\mathcal{K}$ and that have minimum and maximum height among all possible search trees.

Tree with minimum height:	Tree with maximum height:

/ 1 P c) Insert the key 2 into the following AVL tree. *After that*, remove the key 14 from the resulting tree.



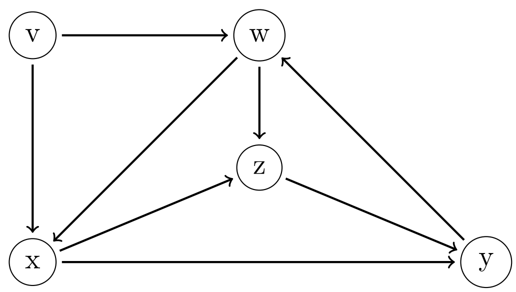
After insertion of 2:	After deletion of 14:

/ 2 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.

An inorder traversal of a binary search tree creates a sorted list of the stored keys.	☐ True	☐ False
If a sequence of m operations has a worst case running time of $\mathcal{O}(m)$, then each single operation has a worst case running time of $\mathcal{O}(1)$.	☐ True	☐ False
Selection sort is a stable sorting algorithm.	☐ True	☐ False
On an array that is completely preordered, the running time of selection sort is linear.	☐ True	☐ False

/ **1 P**

e) Given the following graph $G = (V, E)$, provide a set $E' \subset E$ of smallest cardinality, such that $G' = (V, E \setminus E')$ can be sorted topologically. Provide also a topological ordering for G'.



$E' =$ ______________________

topological ordering for G':

____

/ **1 P**

f) Execute one pivoting step of *quicksort* on the following array (in-situ, i.e., without auxiliary array). Use the rightmost element of the array as pivot.

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

1	2	3	4	5	6	7	8	9	10	11

/ **1 P**

g) The following array contains the elements of a min-heap stored in the usual fashion. Specify the array after the minimum has been removed and the heap condition has been reestablished.

2	8	10	9	12	15	11	13	14
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8

/ 3 P

h) Consider the following recursive formula:

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

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 5.

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

Derivation (if required):

Closed and simplified form:

$T(n) = T(5^k) =$

Proof by induction:

/ 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.

```
for(int i = 1; i <= n/2; i += 2)
    for(int j = n; j >= i; j -= 1)
        for(int k = n; k > 2; k /= 2)
            ;
```

/ 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.

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

/ 1 P

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

$$\sqrt{n^3},\ \binom{n}{4},\ n!,\ \frac{n^2}{\log n},\ n\log n,\ 3^n,\ (\log n)^4$$

Solution: ________, ________, ________, ________, ________, ________, ________.

Problem 2. / 6 P

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 3. / 13 P

A company got the job to produce a cable of length at least L. Since many earlier produced cable pieces are available in the warehouse, the company decides to build the requested cable by putting some of these pieces together. There exist n cable pieces where piece i has length l_i. If two cable pieces of lengths l_i and l_j are put together, we obtain a cable of length $l_i + l_j$. To avoid the resulting cable to be unnecessarily long, the goal is to produce a cable whose length is minimum among all possibilities to produce a cable of length $\geq L$. You can assume that the warehouse contains sufficiently many cable pieces, i.e. that the sum of the lengths of the cable pieces is at least L.

Example: We want to produce a cable of length $L = 6$, and the warehouse contains cable pieces with lengths $l_1 = 3$, $l_2 = 4$ and $l_3 = 5$. The best choice of cable pieces is $\{1, 2\}$, because these two cable pieces together have overall length 7. Notice that the choices $\{2, 3\}$ and $\{1, 2, 3\}$ also lead to a cable of length ≥ 6, but they are not optimal since the lengths of the produced cables is 9, or 12, respectively. Thus, we do not want to output them.

/ 9 P

a) Provide a dynamic programming algorithm for computing the minimum length of a cable that can be obtained by putting suitable cable pieces from $\{1, \dots, n\}$ together, and that has overall length at least L. For the example above, the algorithm should return 7. 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 cable length be obtained from the DP table?

Hint: The trivial algorithm that simply inspects all possible solutions does **not** give any points, because it is not a dynamic programming algorithm.

/ 2 P

b) Describe in detail how you can recognize from the DP table which cable piece is contained in an optimal solution.

/ 2 P

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