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
Thomas Tschager

Exam
Datenstrukturen und Algorithmen
D-INFK

August 5, 2015

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 180 minutes to solve the exam.

Good luck!

Student number: ______________________________

problem	1	2	3	4	$\boldsymbol{\Sigma}$
max. score	18	12	9	11	50
$\sum$ score					

Problem 1.

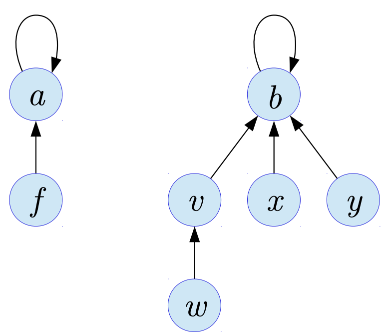
Please note:

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) 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

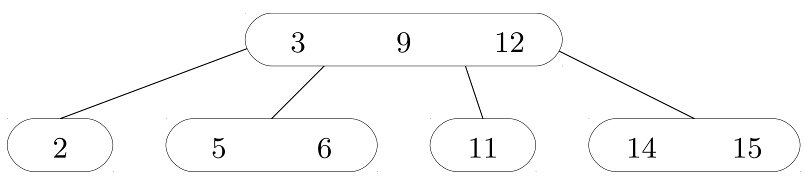
1 P b) On the following union-find data structure, execute first $\textsc{Union}(a, c)$, and after that execute $\textsc{Union}(\textsc{Find}(f), b)$. Use "union by height", and draw the data structure that results after the two operations.



1 P c) 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 d) Insert the key 7 into the following 2-3-4 tree (B tree of order 4). After that, insert the key 8 into the resulting tree. Perform also the necessary structural changes.



After inserting 7:	After inserting 8:

1 P e) 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

2 P f) 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 answers.

Mergesort can be implemented as a stable sorting algorithm.	☐ TRUE	☐ FALSE
In an AVL tree, the number of vertices in the left and in the right subtree may differ by at most 1.	☐ TRUE	☐ FALSE
In a splay tree with n keys, searching for key requires time $\mathcal{O}(\log n)$ in the worst case.	☐ TRUE	☐ FALSE
Let $G = (V, E)$ be a weighted graph. If the minimum spanning tree of G is unique, then G does not have two edges with the same weight.	☐ TRUE	☐ FALSE

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

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

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 $\sum_{i=0}^{k} q^i = \frac{q^{k+1}-1}{q-1}$.

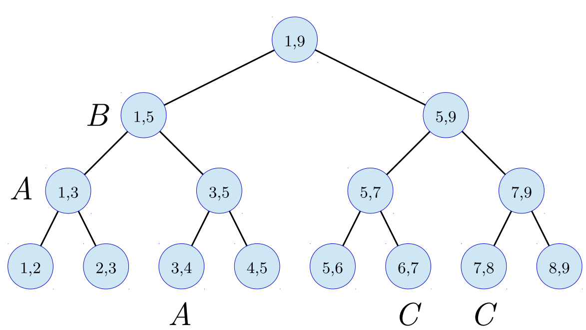
1 P i) Specify (as concisely as possible) the asymptotic running time of the following code fragment in Θ notation depending on $n \in \mathbb{N}$. You do not need to justify your answer.

```
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) Consider the following segment tree that stores the intervals A, B, and C. Draw the resulting tree after the insertion of the interval $D = [1, 8]$. Mark all nodes that are visited in a query for (listing all intervals that contain) $x = 3.7$.



4 P l) A complete ternary search tree is a search tree in which every inner node has exactly three children, and in which every leaf has the same depth h (by definition, the root has depth 0). Deduce a recursive formula depending on h for the number of leaves in a complete ternary tree, and justify your derivation. Resolve the recursion and prove the correctness of your resolution with mathematical induction over h.

Problem 2.

Motivation. In the context of an infrastructure project, charging stations for electrical cars should be build along a highway. It is assumed that an electrical car can cover a distance of 100 km with a single battery charge. There are n possible locations for the charging stations, and an arbitrary large subset of locations for building the charging stations should be selected. However, we do not want to build too many charging stations. Therefore, the locations should be selected such that the distance of two consecutive charging stations is as close as possible to, but not above 100 km.

Problem definition. You are given n possible locations where $d(i)$ is the distance of the i-th location to the starting point of the highway. Moreover we have $d(0) = 0$, and $d(n+1)$ is the total length of the highway. A cost function $c(x) = (100 - x)^2$ is defined for a distance x between two consecutive charging stations. You should select a subset $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ of locations such that an electrical car has to drive at most 100 km to the next charging station (resp. the end of the highway) and such that the total cost of all road sections

$$\sum_{j=0}^{k} c\left(d(i_{j+1}) - d(i_j)\right)$$

with $i_0 = 0$ and $i_{k+1} = n+1$ is minimized. Note that k is not part of the input, i.e. the problem asks for an optimal subset I of *arbitrary* size.

Example: There are $n = 3$ possible locations with $d(1) = 90, d(2) = 100$ and $d(3) = 180$. The total length of the highway is $d(4) = 280$.

While $I = \{2, 3\}$ gives a total cost of 400, the optimal choice is $I^* = \{1, 3\}$ with a total cost of 200.

7 P a) Provide a dynamic programming algorithm for computing the minimal total cost of a selection of locations. 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 total cost be obtained from the DP table?

Hint: The trivial algorithm that simply enumerates all possible solutions does **not** give any points (it is not a dynamic programming algorithm).

2 P b) Describe in detail how you can recognize from the DP table which locations are selected for building the charging stations, such that the total cost is minimized.

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

Problem 3.

Motivation. You want to travel by train from Zürich to Hamburg and there are multiple routes available. You already know the cost of all possible sections of the routes and search for the route with minimum total cost.

Problem definition. You are given a set of stations $V = \{s, t, v_1, \ldots, v_n\}$, where s is your starting point and t is your destination. The other stations v_i are all possible intermediate stops. Moreover, E is the set of directed connections such that $(v, w) \in E$ if and only if there is a connection from station v to station w. Finally, for every connection $(v, w) \in E$ you are given the cost $c(v, w) > 0$. You search for the route from s to t with minimum total cost, i.e. with a minimum sum of the costs of all sections of the route.

2 P a) Name an efficient algorithm for the above problem. Which data structure has to be used in the implementation in order to achieve an efficient running time? Provide the running time in dependency of the number of stations $|V|$ and connections $|E|$.

You have a coupon of value 30 swiss francs, which can be used for a connection with a cost of at least 50 francs. The coupon can only be used once.

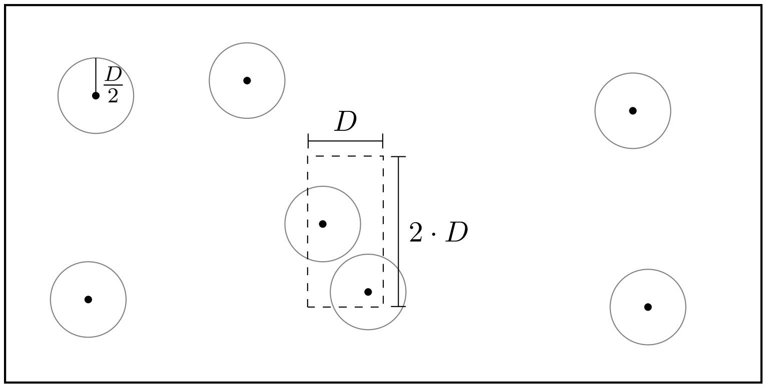
4 P b) Describe the construction of a directed weighted graph $G = (V', E', c')$, such that a shortest path from s to t in G corresponds to a route of minimum total cost achievable with the coupon. Note that the coupon does not necessarily have to be used, because there can be a cheaper route, where the coupon cannot be used on any section. The graph G should be constructed such that the shortest path in G can be computed in the same asymptotic running time as in a).

The train company wants to check the tickets of all passengers. Ticket inspectors should check the tickets on several connections, such that the tickets of all passengers are inspected independent of their choice of the route from s to t. On how many connections the tickets need to be inspected at least, such that every possible route uses at least one connection, where tickets are inspected?

3 P c) Model the above problem as a flow problem. For this purpose, describe the construction of an appropriate network $N = (V'', E'', c'')$ with the vertex set V'' and edge set E'', and describe which capacities c'' the edges should have. Name an efficient algorithm for computing the maximum flow from s to t in N. How can you deduce from the value of a maximum flow on how many connections the tickets need to be inspected at least?

Problem 4.

A company specialized in air traffic control is responsible for monitoring the airspace of Switzerland. It should be guaranteed that two airplanes at the same cruise altitude stay in a minimal pairwise distance of at least D. There are n airplanes at the same altitude, and for every airplane the position (x_i, y_i) is given. An alarm should be set off if two airplanes are in a distance smaller than D.



3 P a) Prove that in a rectangle with side lengths $2 \cdot D$ and D there cannot be more than eight points, if all pairwise distances between the points are at least D.

8 P b) Design an efficient scanline algorithm for the above problem. Address the following aspects in your solution.

1) In which direction is the scanline 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 scanline encounters a new stopping point?
4) Provide the running time of the algorithm in dependency on n and justify your answer.

Remark: For the sake of simplicity we assume that for two arbitrary airplanes i and j we have $x_i \neq x_j$. Note that a trivial algorithm can compute all pairwise distances of the airplanes in time $\mathcal{O}(n^2)$. Therefore, no points are given for an algorithm with quadratic running time in b).