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

August 11, 2011
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	$\mathbf{\Sigma}$
Possible Points	12	7	9	10	9	47
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 $(74 \cdot 93)$, $(51 \cdot 80)$, and $(74 \pm 80) \cdot (51 \pm 93)$.

$x =$ ____________________, $y =$ ____________________

1 P b) Give a sequence of 5 numbers for which bubblesort performs exactly 10 swaps of keys in order to sort the sequence.

Sequence: ________________________

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?

25	60	32	61	62	52	57	80	86
1	2	3	4	5	6	7	8	9

1 P d) Assume that the keys 7, 40, 13, 3, 24, 17, 1, 10, 14, 31, 15, 4, 23, 28, 11 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 binary search tree containing the keys $1, 2, 3, 4, 5, 6, 7, 8$, such that its preorder traversal begins with the sequence $5, 3, 1, 2$ and its postorder traversal ends with the sequence $7, 8, 6, 5$.

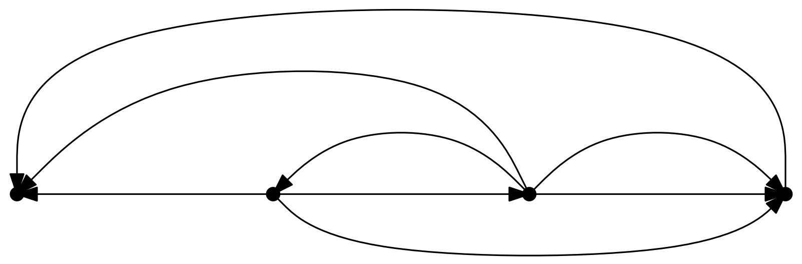
1 P f) Give a sequence of at most 5 access operations to the list $A \to B \to C \to D \to E$, such that exactly 17 comparisons are performed when using the move-to-front rule.

Sequence of keys to access: ____________________

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

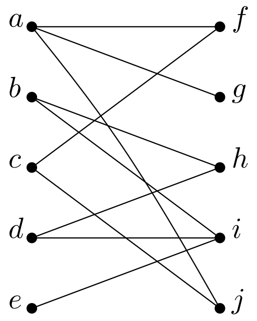
Key	a	b	c	d	e	f	g	h
Frequency count	41	33	70	39	23	25	78	98

1 P h) 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 i) Give an example of a directed graph $G = (V, E)$ that has at most 6 edges and for which the augmenting path algorithm (Ford-Fulkerson) performs up to 10000 flow augmentations if the augmenting paths are chosen in an unfortunate manner. Specify the capacity of each edge. Also label the start node s and the target node t.

1 P j) 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.



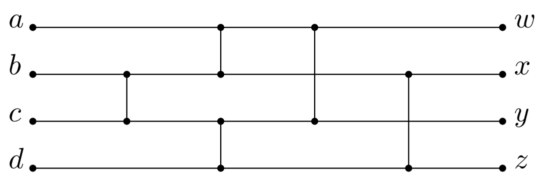
1 P k) Complete the missing parts of the following algorithm, such that the resulting algorithm is a 2-approximation algorithm for the minimum vertex cover problem.

```
**Input**: Graph $G = (V, E)$
$S = \emptyset$;
$U = E$;
**while** $U \neq \emptyset$ **do**
    wähle eine Kante $e = (u, v) \in U$ ;

    $S = S \cup$ ________________________________ ;

    $U = U \setminus$ ________________________________ ;
**end**
**return** $S$;
```

1 P l) The following network of comparators and wires is not a sorting network. Give values for the inputs a, b, c, d of the network which the network fails to sort. (A sorted output is of the form $w \leq x \leq y \leq z$.)



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

- $\sqrt{2^n}$
- $3^{\frac{n}{2}}$
- $\dfrac{2^n}{n!}$
- $n \log(n)$
- $n^{\log n}$
- $n^{\frac{3}{2}}$

3 P b) Consider the following recursive equation:

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

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 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 j := 1 until j > n loop
  from k := 1 until k > n loop
    k := k + j
  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):

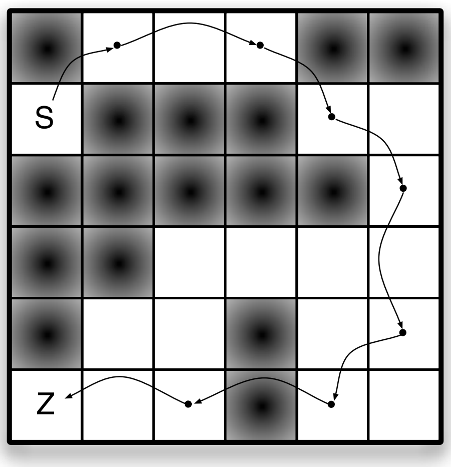
```
from j := 1 until j > n*n loop
  from k := 2 until k > n*n*n loop
    k := k * 2
  end
  j := j + 1
end
```

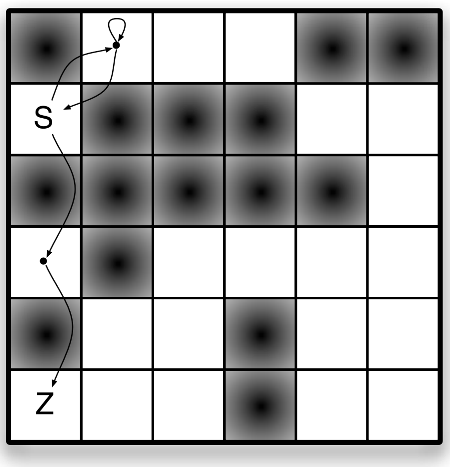
1 P e) Determine the asymptotic running time of the following code, depending on $n \in \mathbb{N}$, expressed in Theta-notation (you don't need to justify your answer):

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

Task 3:

This task is about computing a shortest path for a pogo-stick race on a playing field. A playing field consists of a grid of a times b fields, where each field is either *free* or *blocked* by an obstacle. Using a pogo-stick, one can jump between free fields. Blocked fields cannot be visited, however. We denote the *range* of a jump by (i, j), where $i, j \in \{-3, \ldots, 3\}$, meaning that the movement of the jump is i fields in the direction of x and j fields in the direction of y. Hence, there are 49 diffenrent ranges possible for a jump, from $(-3, -3)$ to $(3, 3)$. Two consecutive jumps must not, however, differ in range by more than 1 per direction. For example, if the last jump had a range of (-2,3), i.e., 2 to the left and 3 upwards, then the next jump is restricted to one of the ranges (-1,2), (-1,3), (-2,2), (-2,3), (-3,2), and (-3,3).
Now, we are looking for the shortest path, measured in the number of jumps, from a given start field s to a given target field z, where both s and z are free. A jump always has to stay within the limits of the playing field. It is possible to jump over blocked fields. We assume that initially, the range of the jump at the field s is (0,0), and that the range of the jump upon arrival at z is arbitrary. In the following examples, the shortest path consists of 8 and 5 jumps, respectively:





2 P a) Model the state space as a directed graph $G = (V, E)$. Describe which nodes and vertices the graph has. Make a precise statement about which nodes are reachable from a given node $v \in V$.

4 P b) Design an algorithm, as efficient as possible, for the problem described above. Briefly describe the algorithm in words. Then, state your algorithm in pseudocode. The output of the algorithm should be the length of a shortest path, or a statement that no path exists from s to z.

1 P c) Specify the running time of your algorithm depending on a and b.

1 P d) Explain how your algorithm can be extended in order to return the computed shortest path, i.e., as a sequence of fields from s to z. It suffices to describe the necessary addiditons to b) in pseudocode.

1 P e) Does G have a topological sort (topological ordering)? Briefly justify your answer.

Task 4:
Let $S = \{p_1, \ldots, p_n\}$ be a set of points in the plane. We are looking for a closest pair of points in that set, i.e., two points which have the smallest Eucledian distance between any pair of points in S. More formally, we are looking for a pair $a, b \in S$, $a \neq b$, such that $\text{dist}(a, b) = \min_{1 \leq i < j \leq n} \text{dist}(p_i, p_j)$, where for a pair of points $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$, the Eucledian distance is defined as $\text{dist}(p_i, p_j) := \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.
In this task you will design a sweep line algorithm that computes, as efficiently as possible, a closest pair of points of a given set S.

1 P a) Specify the halting points (stopping points) of the sweep line and the order in which they are encountered by the sweep line.

1 P b) Specify the invariant of the sweep line, i.e., which information is stored in the data structure D associated with the sweep line.

1 P c) Which data structure D is a good choice in order to efficiently implement your algorithm?

4 P d) Describe your algorithm in pseudocode.

3 P e) Specify the running time of your algorithm. To that end, derive a constant upper bound on the number of necessary distance computations that are performed at a halting point of the sweep line. Further, specify the running time of the operations that are performed on the data structure D during the execution of your algorithm, along with a brief justification.

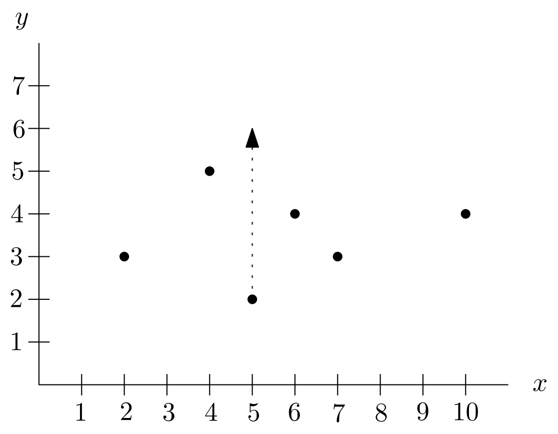
Note: You may assume that no two points of S have the same x- or y-coordinate.

Task 5:

In this task you will design a data structure for the following problem: Let $p_1, p_2, \ldots, p_n$ be n points with integer coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. The x_i coordinates are all distinct and immutable. The points may, however, be shifted along the y-axis, i.e., the values y_i may be changed by certain operations.
The data structure that is to be designed has to support the following operations as efficiently as possible:

- `RangeMin(`$x_{\min}$`, `$x_{\max}$`)`:
  Return the smalles y-coordinate of a point lying in the interval $[x_{\min}, x_{\max}]$.
- `Update(`x_i`, `y_{new}`)`:
  Set the y-Koordinate of point (x_i, y_i) to the value y_{new}.

Hence, in the example below the result of `RangeMin(4, 7)` would be the value 2. After executing the operation `Update(5, 6)`, `RangeMin(4, 7)` would return the value 3.



3 P a) Design and describe a data structure for the problem descibed above. Specify a helpful invariant, which should hold for the values stored in your data structure (independent of the number of performed `Update` operations).

3 P b) Describe how the data structure of part a) can be used in order to efficiently implement the operation `RangeMin(`$x_{\min}$`, `$x_{\max}$`)`. Further, specify the running time of this operation in the worst case.

3 P c) Describe how the data structure is updated when the operation `Update(`x_i`, `y_{new}`)` is executed.