Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chintan Shah

Wintersemester 2012
Abschlussklausur
20th February, 2013

Effiziente Algorithmen und Datenstrukturen I

Last Name	First Name	Matrikel No.
....	....	....

Hall	Seat No.	Signature
....	....	....

General Information for the Examination

- Write your name and matrikel no. on all extra supplementaries (Blätter) provided.
- Please keep your identity card readily available.
- Do not use pencils. Do not write in red or green ink.
- You are not allowed to use any device other than your pens and a single sided handwritten A4 sized paper (with your name clearly written on top).
- Verify that you have received 12 printed sides (check page numbers).
- Attempt all questions. You have 180 minutes to answer the questions.

Left Examination Hall from ...... to ...... / from ...... to ......
Submitted Early at ......
Special Notes:

	A1	A2	A3	A4	A5	A6	A7	A8	A9	Σ	Examiner
Maximum	10	2	2	3	4	4	4	5	6	40	
1st Correction											
2nd Correction											

Aufgabe 1 (10 Points)

For each of the following statements, mark whether it is true or false. Example:

1. $2 + 2 = 4$. ☒ True ☐ False

You receive 1 point for a correct answer, 0 points for not attempting and -1 point for an incorrect answer. The total points you receive for this question (out of 10) will be greater than or equal to 0.

1. $T(n) = 2T(\frac{n}{4}) + \sqrt{n} \Rightarrow T(n) \in o(\sqrt{n}\log^2 n)$. ☒ True ☐ False
2. The length of every path from the root to a leaf in an AVL tree with n nodes is $\Theta(\log n)$. ☒ True ☐ False
3. Every class of hash functions $\mathcal{H}$ which is universal, is also pairwise independent. ☐ True ☒ False
4. There exists a class of hash functions $\mathcal{H}$ which is $(3, 3)$-independent, but not 3-independent. ☒ True ☐ False
5. In a binomial heap, the DELETE-MIN operation can be performed in constant time. ☐ True ☒ False
6. Using Fibonacci Heaps, Dijkstra's algorithm runs in time $O(|V|\log(|V|) + |E|)$. ☒ True ☐ False
7. In the Union Find data structure with path compression, the rank of a node could be equal to the rank of its parent. ☐ True ☒ False
8. When implementing Union Find with path compression, the amortized running time for the FIND operation is $o(\log n)$. ☒ True ☐ False
9. The value of any feasible flow is less than or equal to the value of any cut in a graph. ☒ True ☐ False
10. Every flow is also a preflow. ☒ True ☐ False

Aufgabe 2 (2 Points)

In the following, we consider an array implementation of a binary heap. The indexing starts from 1.
Assuming all elements are distinct, give the list of all indices in a heap of size 15 where

(a) the 3rd smallest element can appear.

(b) the 5th smallest element can appear.

Aufgabe 3 (2 Points)

Let f be a flow in a network, and let α be a real number. The scalar flow product, denoted $\alpha \cdot f$, is a function from $V \times V$ to $\mathbb{R}$ defined by

$$(\alpha \cdot f)(u, v) = \alpha \cdot f(u, v)$$

If f_1 and f_2 are flows, show (formally) that $\alpha f_1 + (1 - \alpha) f_2$ is also a flow for $0 \leq \alpha \leq 1$.

Aufgabe 4 (3 Points)

Give tight asymptotic lower and upper bounds for the following recurrence relation:

$$T(n) = T(\sqrt{n}) + 1$$

where $T(2) = 1$.

Aufgabe 5 (4 Points)

For any positive integer n, show a sequence of Fibonacci heap operations that creates a Fibonacci heap consisting of just one tree that is a linear chain of n nodes.

Aufgabe 6 (4 Points)

Consider the mincost flow problem

$$
\begin{array}{rll}
\min & \sum_{e\in E} c(e)f(e) & \\
\text{s.t.} & \ell(e) \le f(e) \le u(e), & \forall e \in E \\
& a(v) \le f(v) \le b(v), & \forall v \in V
\end{array}
$$

where $a : V \to \mathbb{R}$, $b : V \to \mathbb{R}$, $\ell : E \to \mathbb{R} \cup \{-\infty\}$, $u : E \to \mathbb{R} \cup \{\infty\}$ and $c : E \to \mathbb{R}$. Show that if the flow problem is feasible, then $\sum_{v\in V} a(v) \le 0$.

Aufgabe 7 (4 Points)

Consider a binary heap H implemented with a binary tree data structure (as implemented in the lectures) containing n items. Design an algorithm to find the k-th smallest item in H in $O(k \log k)$ time.

Aufgabe 8 (5 Points)

Design a data structure that handles the database of a single bank account. This data structure should support insertion of past and future transactions, as well as deletion of existing transactions. Assume that no 2 transactions occur at the same date. You should support the following functions:

(a) INIT() : Initializes the account. The initial balance in the account is 0 €.

(b) INS-TRANS(sum,date) : Insert a given transaction at a given date. The sum can be either positive or negative, and should be added to the balance in the account starting from the following day. Note that the date can be arbitrary (not necessarily today's date).

(c) DEL-TRANS(date) : Delete the transaction that occurs at the given date, if there is any. When a transaction is deleted, the corresponding sum should be subtracted from the balance in the account starting from the following day.

(d) BALANCE(date) : Returns the balance in the account at the beginning of the given date.

You should be able to perform INIT() in $O(1)$ time and all other operations in $O(\log n)$ time, where n is the number of transactions.

Aufgabe 9 (6 Points)

The Mathematics and Computer Science department has n faculty members $f_1, f_2, \ldots, f_n$ who will offer n courses $c_1, c_2, \ldots, c_n$ in the coming semester and each faculty member will teach exactly one course. Each faculty member chooses two courses he (or she) would like to teach, and ranks them according to his (or her) preference (rank 1 indicates higher preference and rank 2 indicates lower preference).

(a) We say that a course assignment is a *feasible* assignment if every faculty member teaches a course within his (or her) preference list. How would you efficiently determine whether the department can find a feasible assignment? (2 points)

(b) We say that a feasible assignment is an *optimal assignment* if it maximizes the number of faculty members assigned to their most preferred course. Suggest an efficient algorithm for determining an optimal assignment and analyze its complexity. (4 points)

ROUGH WORK