Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Emo Welzl, Thomas Holenstein, Ueli Maurer, Angelika Steger, Peter Widmayer

February 8, 2011

Algorithms, Probability, and Computing *Fall 2010 Exam*

Candidate:

First name: ....

Last name: ....

Student ID (Legi) Nr.: ....

I attest with my signature that I was able to take the exam under regular conditions and that I have read and understood the general remarks below.

Signature: ....

General remarks and instructions:

1. You can solve the 7 exercises in any order. **You should not be worried if you cannot solve all the exercises!** The best grade will be awarded for significantly less than the maximum 180 points. Nevertheless you should not spend too much time on a single exercise.
2. Check your exam documents for completeness (1 cover sheet and 2 sheets containing 7 exercises).
3. Immediately inform an assistant in case you are not able to take the exam under regular conditions. Later complaints are not accepted.
4. Make sure to write your **student-ID**-number on **all** additional sheets, and your name **only** on this first cover sheet.
5. No auxiliary material allowed.
6. Attempts to cheat/defraud lead to immediate exclusion from the exam and can have judicial consequences.
7. Pencils are not allowed. Pencil-written solutions will not be reviewed.
8. Provide only one solution to each exercise. Cancel invalid solutions clearly.
9. **All solutions must be understandable and well-founded. Write down the important thoughts in clear sentences and keywords. No points will be awarded for unfounded or incomprehensible solutions (except in the multiple-choice parts).**

You can write your solution in English or German.

Good luck!

	achieved points (maximum)	reviewer's signature
1	(30)	
2	(25)	
3	(25)	
4	(25)	
5	(25)	
6	(25)	
7	(25)	
Σ	(180)	

Exercise 1

Consider the following claims and mark the corresponding boxes. Grading: 2 points for each correct marking, 5 points for a correct marking with a correct short justification (or reference), and -2 points for a wrongly marked box (you will receive non-negative total points in any case).

(a) [5 points] The expected number of necessary rotations for an insertion or deletion in a treap is always bounded by a constant, independent of the size of the treap.

[ ] True [ ] False

Justification: ....

....

(b) [5 points] A set P of n points in the plane can be pre-processed in time $O(n \log n)$ such that we can decide for any query point q if it lies inside $\mathsf{conv}(P)$ in time $O(\log n)$.

[ ] True [ ] False

Justification: ....

....

(c) [5 points] For any network and any flow of value $v \in \mathbf{N}$, there exists an s-t-cut of the same capacity v.

[ ] True [ ] False

Justification: ....

....

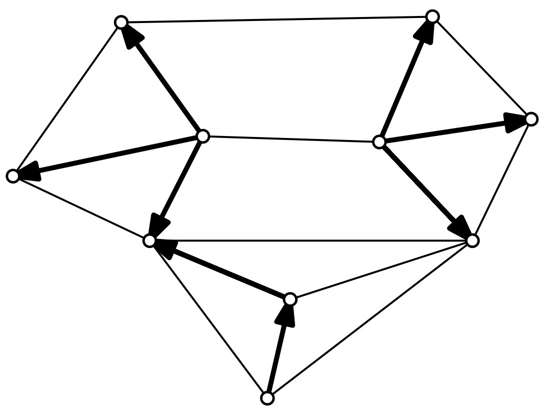
(d) [5 points] An edge contraction in a graph G with at least 3 vertices (as performed in the randomized MinCut algorithm) can decrease the size of a minimum cut of G.

[ ] True [ ] False

Justification: ....

....

(e) [10 points] Complete the following partial orientation (some edges are already oriented) to a Pfaffian orientation.



Exercise 2

(a) [13 points] Solve the following recurrence for $n \in \mathbf{N}_0$.

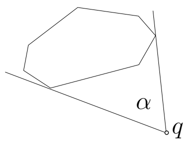
$$a_n = \begin{cases} \frac{3}{n} \sum_{i=0}^{n-1} a_i & \text{for } n \geq 1, \\ 2 & \text{for } n = 0. \end{cases}$$

(b) [12 points] In a random search tree on n nodes, let R_n denote the number of nodes whose right subtree consists of exactly a single node (i.e. they have a single right child). Compute $r_n := \mathbf{E}[R_n]$.

Exercise 3

As a 2-d game programmer, you are given a polygonal spaceship $\mathsf{conv}(P)$ determined by n points $P = \{p_1, \ldots, p_n\} \subset \mathbf{R}^2$ in convex position, given in counter-clockwise order.

(a) [10 points] Describe an algorithm and data structure that reports the angle α under which the spaceship is visible from a query point q in time $O(\log n)$. We assume that $q \notin \mathsf{conv}(P)$.



(b) [15 points] Consider now a second spaceship, given by points $Q = \{q_1, \ldots, q_n\} \subset \mathbf{R}^2$ (again in convex position and counter-clockwise order), and initially $\mathsf{conv}(P) \cap \mathsf{conv}(Q) = \emptyset$. Now each of the spaceships moves linearly, where $\mathsf{conv}(P)$ moves with velocity vector $v_P \in \mathbf{R}^2$ (so that $p_i(t) = p_i + t \cdot v_P \; \forall i$), and $\mathsf{conv}(Q)$ moves with velocity vector $v_Q \in \mathbf{R}^2$.

Describe a flight control algorithm running in total time $O(n \log n)$, reporting "safe" if the spaceships never collide, or otherwise reporting the exact time t when the spaceships will first collide.

Exercise 4

Consider the directed n-dimensional cube Q_n for $n \in \mathbf{N}$, which can be defined as follows: The vertices of Q_n are the vectors from $\{0,1\}^n$, and two vertices are adjacent if they differ by exactly one coordinate; the arc is oriented from the vertex with 0 as at this coordinate to the vector with 1 at this coordinate. Or, more formally: $Q_n = (V, E)$ with $V = \{0,1\}^n$ and

$$E = \{(x, y) \mid \exists k : x_i = y_i \text{ for } i \neq k \text{ and } x_k = 0, y_k = 1\}$$

(a) [5 points] Consider the network $(Q_3, (0,0,0), (1,1,1), c)$ with unit capacities $c(e) = 1 \; \forall e \in E$. Find a maximum flow in this network, which uses every edge with strictly positive flow. Explain why the flow is maximum.

(b) [2 points] Consider the same network as in part a). Is there is an integral flow of the same (maximum) value?

(c) [15 points] One can partition the vertices of a hypercube into levels by the number of ones that the vectors contain: For $k \in \{1, \ldots, n\}$, we call the *k-level* of a hypercube the set consisting of the $\binom{n}{k}$ vertices composed from k ones and $n - k$ zeros. Prove the following statement:

> For $k \in \mathbf{N}, k < \frac{n}{2}$, there are at least $\binom{n}{k}$ edge-disjoint paths in Q_n that lead from the k-level to the $(n-k)$-level.

(d) [3 points] Can you also prove the statement if the paths are required to be vertex-disjoint?

Exercise 5

(a) [6 points] State the definition of a ρ-reduction of a computational problem $\mathcal{C}$ to $\mathcal{C}'$ (for a function $\rho : \mathbf{R} \to \mathbf{R}$), or in other words $\mathcal{C} \preceq_{\rho}^{R} \mathcal{C}'$.

(b) [12 points] Assume that $\mathcal{C} \preceq_{\rho}^{R} \mathcal{C}'$ for two computational problems $\mathcal{C}, \mathcal{C}'$.

Recall that the performance-complexity function is defined as $\overline{\pi}^{\mathcal{C}}(c) := \sup_{A:\gamma(A)\leq c} \pi(A, \mathcal{C})$ for a complexity function γ. Furthermore γ_R being a complexity function for R means that $\forall A\ \gamma(RA) \leq \gamma_R(\gamma(A))$.

Prove that if $\overline{\pi}^{\mathcal{C}}(\gamma_R(c)) < \rho(\alpha)$, then $\overline{\pi}^{\mathcal{C}'}(c) \leq \alpha$.

(c) [7 points] Show that the discrete logarithm problem is random self-reducible.

Hint: Random instances?

Exercise 6

If two Boolean matrices (matrices with entries 0 or 1) A and B have the same number of rows, they can be concatenated horizontally to form the matrix $A \boxbar B := [A\ B]$. If A and B have the same number of columns, they can be concatenated vertically which we denote by $A \boxminus B := \begin{bmatrix} A \\ B \end{bmatrix}$.

A *matrix building program* (*MBP*) is a list of rules of the form "$X \leftarrow Y \boxbar Z$" or "$X \leftarrow Y \boxminus Z$", where X is a *variable* and Y and Z are either variables assigned earlier or elementary matrices [0] or [1]. Each program is only passed once sequentially from top to bottom.

Let us make a small example. The MBP $\mathcal{P}$:

$$X \leftarrow [0] \boxminus [1]$$

$$Y \leftarrow X \boxbar X$$

produces the matrix

$$M(\mathcal{P}) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

Let us say that the *length* $|\mathcal{P}|$ of an MBP $\mathcal{P}$ is the number of rules (lines) it has and that $M(\mathcal{P})$ always denotes the matrix produced by the last rule in $\mathcal{P}$ (after the entire program has been processed).

(a) [5 points] Exhibit a sample MBP $\mathcal{P}$ with n lines such that the matrix $M(\mathcal{P})$ which it produces has exponentially many rows and exponentially many columns (in n).

(b) [5 points] Describe an algorithm that takes an MBP $\mathcal{P}$ and computes the number m of rows and the number n of columns of the matrix $M(\mathcal{P})$ in time polynomial in the length of $\mathcal{P}$.

(c) [15 points] Describe a randomized algorithm, running in time polynomial in $n = |\mathcal{P}|+|\mathcal{Q}|$, that takes two MBPs $\mathcal{P}$ and $\mathcal{Q}$ as input and tests whether they produce the same matrix. If $M(\mathcal{P}) = M(\mathcal{Q})$, your algorithm has to output 'yes' always, if $M(\mathcal{P}) \neq M(\mathcal{Q})$, it has to output 'no' with probability at least 1/2.

Hint: A Boolean matrix $M \in \{0,1\}^{m\times k}$ can be represented by the bivariate polynomial

$$p_M(x,y) := \sum_{i=1}^{m} \sum_{j=1}^{k} M_{ij}\, x^i y^j,$$

so that for example $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ would be represented as $axy + bxy^2 + cx^2y + dx^2y^2$. You can interpret $p_M(x,y)$ as a polynomial over any field $\mathbb{F}_q$, where q is a prime of your liking. Don't worry about finding large enough primes! This is known to be possible and you can simply assume an oracle giving you whatever prime you need.

Exercise 7

(a) [10 points] Prove the following statement: There exists a constant $\rho > 0$ such that for any language $L \in \mathrm{NP}$, there exists a polynomial time verifier $V(x, w)$ with the following properties. The verifier expects a proof w of size polynomial in $|x|$ for the statement $x \in L$. It first reads x, tosses $\mathcal{O}(\log |x|)$ random coins, reads 3 bits of w, then accepts or rejects. If $x \in L$, then there exists a proof w such that the verifier accepts with probability 1. If $x \notin L$, then for all w, the verifier rejects with probability at least ρ.

Hint: You may assume any of the theorems stated in the lecture notes (even the ones that were too hard for us to prove during the lecture).

(b) [15 points] Somebody claims that there exists a constant $\rho > 0$ such that for any for any $L \in NP$, there exists a polynomial time verifier $V(x, w)$ with the following properties. The verifier expects a proof w of size polynomial in $|x|$ for the statement $x \in L$. It first reads x, tosses $\mathcal{O}(\log |x|)$ random coins, reads 3 consecutive bits (i.e., a substring $w_i w_{i+1} w_{i+2}$ for some i) of w, then accepts or rejects. If $x \in L$, then there exists a proof w such that the verifier accepts with probability 1. If $x \notin L$, then for all w, the verifier rejects with probability at least ρ. Prove that if that somebody is right, then $P = NP$.

Hint: Reduce the problem to a satisfiability instance and make use of its specific properties.