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

2/08/2010

Algorithms, Probability, and Computing — *Fall 09 Exam*

Candidate:

First name: .....

Last name: .....

Registration Number: (Stud.-Nr.) .....

I confirm with my signature that I 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 6 assignments in any order. You should not be worried if you cannot solve all exercises (60 out of 72 points are sufficient for achieving the best grade).
2. Check your exam documents for completeness (1 title sheet and 1 sheet containing 6 assignments).
3. Immediately inform an assistant in case you are not able to take the exam under regular conditions. Later complaints will not be accepted.
4. Pencils are not allowed. Pencil-written solutions will not be reviewed.
5. No additives (Hilfsmittel) allowed.
6. Attempts to defraud yield to immediate exclusion from the exam and can have judicial consequences.
7. Provide only one solution to each exercise. Cancel invalid solutions clearly.
8. **All solutions must be understandable and well-founded. Write down the important thoughts in articulative sentences and keywords. Unfounded or incomprehensible solutions will not be awarded.**

   You can write your solution in English or German.
9. Make sure to write your name on all the sheets.

Good luck!

	achieved points (maximum)	reviewer's signature
1	(12)	
2	(12)	
3	(12)	
4	(12)	
5	(12)	
6	(12)	
Σ	(72)	

Problem 1

For $n \geq 2$, consider a random search tree on the elements $\{1, 2, \ldots, n\}$ (according to the distribution from the course). We define an indicator variable L_i that is 1 if element i is a leaf in the tree and 0 otherwise.

1. Determine $\mathbf{E}[L_i]$ for $1 \leq i \leq n$.
2. What is the expected number of leaves in a random search tree on n elements?
3. Compute $\mathbf{E}[L_i \cdot L_{i+1}]$ for $1 \leq i \leq n-1$, $\mathbf{E}[L_i \cdot L_{i+2}]$ for $1 \leq i \leq n-2$ and $\mathbf{E}[L_i \cdot L_{i+3}]$ for $1 \leq i \leq n-3$.

All answers should come with a proof.

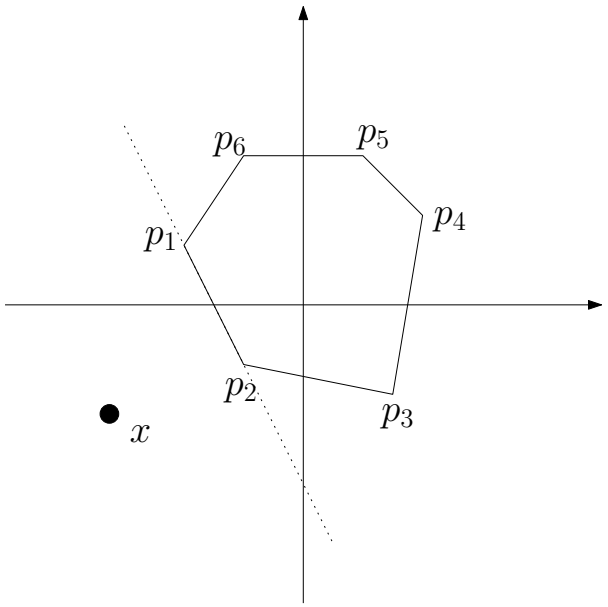

Figure 1: Point x sees edge 1 and edge 2, but does not see any of the other edges.

Problem 2

You are given a convex polygon with vertex set $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$. The points $p_1, \ldots, p_n$ are in counterclockwise order, and you can assume that the origin lies in the interior of the polygon. The polygon has n edges. For $1 \leq i \leq n-1$, edge i is the line segment $p_i p_{i+1}$, and edge n is the segment $p_n p_1$. For a point x outside the polygon, we say x *sees* edge i of the polygon if x and the polygon lie on different sides of the line through edge i (See Figure 1). Design a data structure of linear size and a query algorithm answering queries of the following types:

1. Given a point x outside the polygon and an index $i \in \{1, \ldots, n\}$, decide in *constant* time whether x sees edge i.
2. Given a point x outside the polygon, find indices i and j in $O(\log n)$ time such that x sees edge i, but does not see edge j.
3. Given a point x outside the polygon, compute the number of edges of the polygon that x can see in $O(\log n)$ time.

In all cases you can assume that x does not lie on a common line with any edge of the polygon.

Problem 3

1. We consider the shortest-path augmentation algorithm from the lecture. It iteratively finds a shortest s–t-path P in the residual network G_f and augments the flow f along P. In the lecture, we proved that this algorithm terminates after $O(mn)$ augmentations (here, n is the number of vertices and m is the number of edges). Sketch the proof of this statement. In particular, prove why the length of a shortest s–t-path cannot decrease when augmenting along such a path.

2. Consider a network (G, s, t, c). Let P be an s–t-path. We define

$$c(P) := \min\{c(e) \mid e \text{ is an edge in } P\} .$$

and

$$\gamma(G) := \max\{c(P) \mid P \text{ is an } s\text{–}t\text{-path}\} .$$

Suppose G has no cycles of length 2, and let P be any s–t-path. We start with the 0-flow that has value 0 on all edges, and augment this flow along the path P. Let N' denote the residual network. Show that $\gamma(N') \leq \gamma(N)$. In words, show that augmenting along an s–t-path cannot increase the capacity of a maximum capacity s–t-path.

Problem 4

Consider Karger's algorithm for finding a minimum cut in an undirected graph. It works by iteratively contracting a randomly chosen edge, until only two vertices (and thus only one cut) are left.

1. Describe Karger's algorithm in more detail and analyze its success probability.
2. Suppose I give you a randomized algorithm called `ME` that takes as input an undirected graph G, runs in linear time, and outputs a number k with the following property: With probability at least $1/n$, the number k is in fact the size of a minimum cut of G. Can you turn this into an algorithm of running time $O(n^2)$ and with constant probability returns the true size of a minimum cut? What is the difficulty? What is the crucial difference between Karger's algorithm and `ME`?

Problem 5

Consider the polynomial $p(x,y) = \sum_{i=0}^{k} \sum_{j=0}^{\ell} a_{ij} x^i y^j$ over the field $GF(q)$. We assume that $q \geq (k+1)(\ell+1)$. The polynomial $p(x,y)$ is not given explicitly (i.e., we do not know the coefficients a_{ij}), but as an oracle. We can send a pair $(x,y) \in GF(q)^2$ to the oracle, which will respond with one element in $GF(q)$, namely the value $p(x,y)$.

1. We want to find out the coefficients a_{ij}. Show how to do this with querying the oracle at most $(k+1)(\ell+1)$ times. Hint: You may assume that you can interpolate a univariate polynomial R of degree d if you know the values $R(x_0), R(x_1), \ldots, R(x_d)$ for $d+1$ distinct values $x_0, \ldots, x_d$ from $GF(q)$, provided that $q \geq d+1$.

2. A wizard claims to know the a_{ij}. You are willing to pay a decent amount for finding out, but you want to make sure that the magician is not lying, i.e., make sure that the coefficients b_{ij} he tells you are really the a_{ij} of your polynomial. Show how you can test whether he is telling the truth by just *one* query to the oracle, with a small error probability. What is your probability to catch the wizard if he is lying? What is your probability of wrongly accusing him of lying when indeed he is telling the truth?

Problem 6

You are given a linear function $f : GF(2)^n \to GF(2)$, and once again, f is not given explicitly, but as an oracle. For given vectors $\vec{a}_1, \ldots, \vec{a}_k \in GF(2)^n$ and values $b_1, \ldots, b_k \in GF(2)$, you want to check whether the following system of linear equations is satisfied:

$$\begin{aligned} f(\vec{a}_1) &= b_1 \\ f(\vec{a}_2) &= b_2 \\ &\vdots \\ f(\vec{a}_k) &= b_k \end{aligned}$$

Needless to say, you do not have the time to query the oracle for each of the k vectors $\vec{a}_i$. You want to perform a randomized check that queries the oracle three times. Your test should always answer "yes" if the k equations are satisfied, and should answer "no" with probability at least $7/8$ if at least one equation is violated. Describe your test in detail and prove the claimed bound on its error probability.