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

November 9, 2008

Algorithms, Probability, and Computing *Fall 09*
Mid-Term Exam

Candidate:

First name: ....

Last name: ....

Registration Number: ....

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 5 exercises in any order. **You should not be worried if you cannot solve all the exercises** (best grade with at least 48 points, i.e. 4 exercises).
2. Check your exam documents for completeness (1 cover sheet and 3 sheets containing 5 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. Pencils are not allowed. Pencil-written solutions will not be reviewed.
5. No additives 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)	
Σ	(60)	

Exercise 1

We consider a random search tree for the elements $\{1, \ldots, n\}$ for $n \geq 3$. Let $p_n^{(1)}$ denote the probability that the descendants of element 1 are exactly the elements 1, 2 and 3, i.e. no further elements are descendants of 1.

(a) Compute $p_n^{(1)}$ for $n = 3$ and $n = 4$.

(b) Set up a recurrence for $p_n^{(1)}$ and solve the recurrence. Remark: Do not be tempted to solve (b) by first solving (c). You *have to* solve (b) via the recurrence method.

(c) For $n \geq 5$ and $2 \leq i \leq n-3$, consider the indicator variable

$$E_i := [\text{the descendants of } i \text{ are exactly } i, i+1, \text{ and } i+2] \ .$$

Express E_i in terms of the indicator variables A_j^i from the lecture notes. These where defined as

$$A_j^i := [i \text{ is an ancestor of } j] \ .$$

(d) Compute $\mathbb{E}[E_i]$. Warning: the events A_j^i you used for solving (c) are most likely *not* independent.

Exercise 2

You are given five elements with the following keys and priorities:

key	priority
1	0.9
2	0.3
4	0.6
5	0.2
6	0.5

(a) Draw the treap according to the given keys and priorities. Reminder: we consider min-heaps, i.e. the root is the element with the smallest priority (i.e., here the element with key 5 and priority 0.2).

(b) Insert the element 3 with priority 0.4 into this treap: draw the treap after insertion, but before any rotation (i.e., here, the element with key 3 is a leaf). Then draw the treap after each rotation, until 3 is at its final position.

(c) Suppose we insert the element with key 3 not with priority 0.4, but with priority $p \in_{\text{u.a.r.}} [0, 1]$. What is the probability that after insertion and all subsequent rotations, its depth is 1? Justify your answer.

Exercise 3

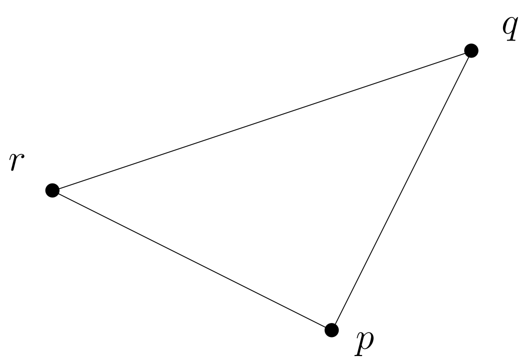
Consider the triangle $T := \mathrm{conv}(\{p, q, r\})$ in the picture below and the three dual lines p^*, q^*, and r^*. Recall that for a point $p = (a, b)$, its dual p^* is the line $ax - b$, and the dual of a line $\ell : ax + b$ is the point $\ell^* = (a, -b)$. We call a line *nice* if it contains at least one of the three vertices p, q, and r, but also contains at least one more point from T. For example, the horizontal line through r is nice, and so is the line connecting p and q. The horizontal line through q is *not* nice.

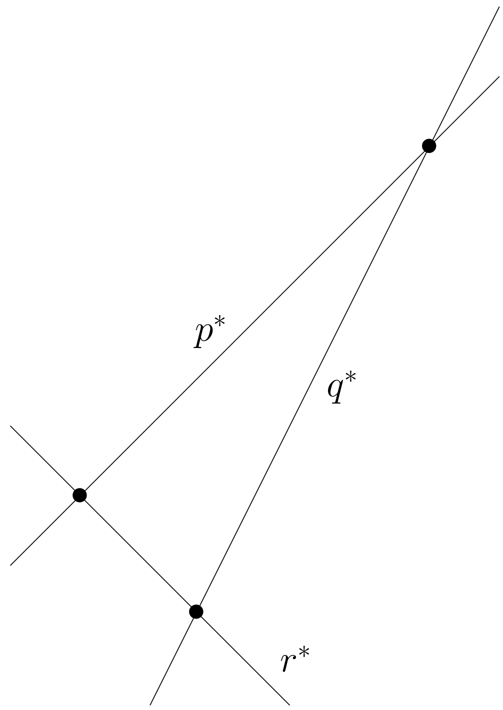
(a) Draw a line into the left picture below that is not nice and intersects the interior of the triangle.

(b) Draw the set of duals of all nice lines, i.e. the set

$$\{\ell^* \mid \ell \text{ is a nice line}\} ,$$

and give a short explanation of your answer.





Exercise 4

You are given a convex polygon C with vertices $p_1, \ldots, p_n$. We define a *scaled version* of C as follows: For $\lambda \in \mathbb{R}_0^+$, we define the set

$$\lambda C := \text{conv}(\{\lambda p_1, \ldots, \lambda p_n\}) ,$$

where λp_i simply denotes scalar multiplication in $\mathbb{R}^2$. We are now interested in the following problem: Given a point $q \in \mathbb{R}$, find the smallest $\lambda \in \mathbb{R}_0^+$ such that $q \in \lambda C$. We call that value $\lambda_C(q)$. Note that such a λ does not always exist.

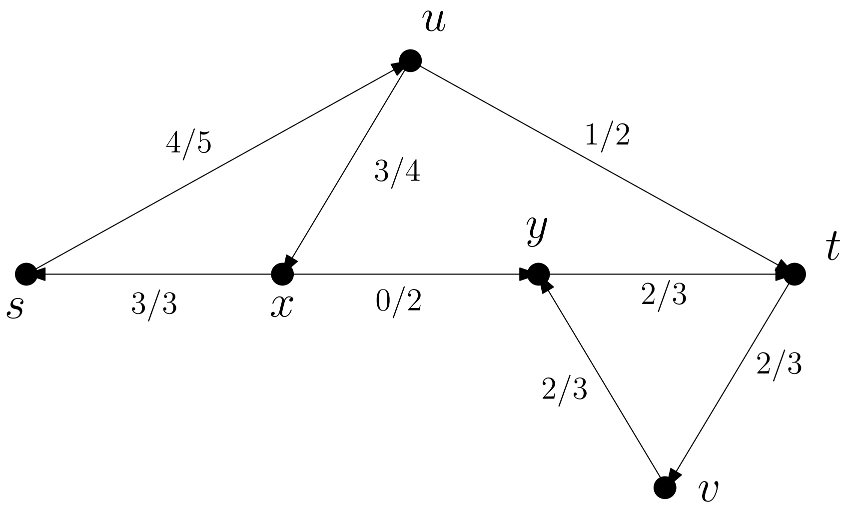
(a) Draw an example of a polygon C and a point q such that there is no $\lambda \in \mathbb{R}_0^+$ such that $q \in \lambda C$. You probably want to draw x-axis and y-axis as well.

(b) Draw an example of a polygon C and a point q where $q \notin C$ and $\lambda_C(q) < 1$.

(c) Design a preprocessing/query algorithm doing the following: On query q, it computes $\lambda_C(q)$ in time $O(\log n)$, outputs `impossible` if no such λ exists. Your data structure should use no more than $O(n)$ space. You can assume that $p_1, \ldots, p_n$ describe the vertices of the polygon in counter-clockwise order. For simplicity, you may assume that the origin $(0, 0)$ does not lie on the boundary of C, and that no edge of C lies on a line with the origin.

Exercise 5

Consider the following network and its flow. Please note that the numbers 4/5 and so on are not fractions: the 4/5 on the edge from s to u does not read "four fifths", but indicates that this is an edge of capacity 5, and the current flow at this edge has value 4.

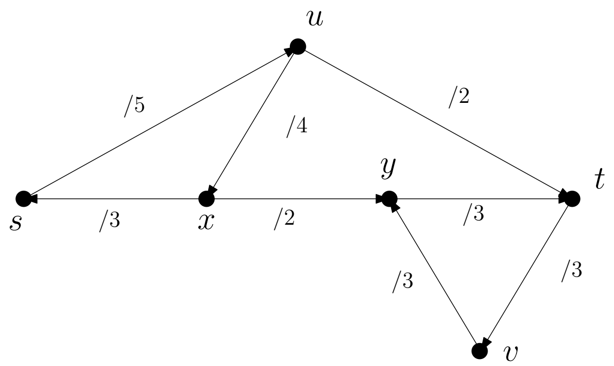


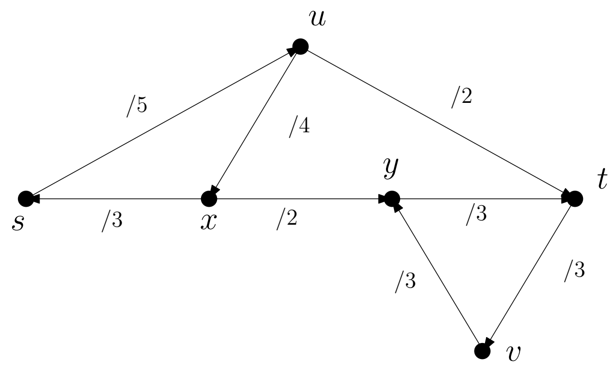
(a) What is the value of the flow?

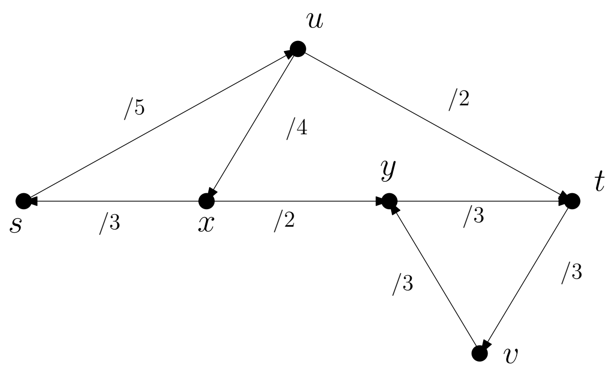
(b) Draw the residual network.

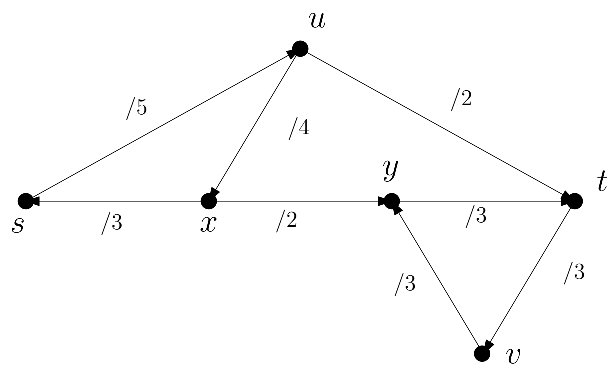
(c) Simulate Ford-Fulkerson on this network: Perform several augemtation steps, starting with the flow drawn above, until you find a maximum flow. In each step, clearly say which augmenting path you are using, and draw the flow after augmenting. For this, you can use the copies of the network that are printed below by filling in the flow values.

(d) Give a min cut of this network, i.e. a set S of vertices containing s but not t minimizing cap(S).


u
/5
/4
/2
y
t
s
/3
x
/2
/3
/3
/3
v


u
/5
/4
/2
y
t
s
/3
x
/2
/3
/3
/3
v


u
/5
/4
/2
y
t
s
/3
x
/2
/3
/3
/3
v


u
/5
/4
/2
y
t
s
/3
x
/2
/3
/3
/3
v