ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Bernd Gärtner, Rasmus Kyng, Angelika Steger, David Steurer, Emo Welzl

Algorithms, Probability, and Computing — Midterm Exam — HS23

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. Check your exam documents for completeness (8 two-sided pages with 5 exercises).
2. You have **2 hours** to solve the exercises.
3. You can solve the exercises in any order. You should not be worried if you cannot solve all exercises! Not all points are required to get the best grade.
4. Immediately inform an assistant in case you are not able to take the exam under regular conditions. Later complaints will not be accepted.
5. Pencils are not allowed. Pencil-written solutions will not be reviewed.
6. No auxiliary material allowed. All electronic devices must be turned off and are not allowed to be on your desk. We will write the current time on the blackboard every 15 minutes.
7. Attempts to cheat lead to immediate exclusion from the exam and can have judicial consequences.
8. Provide only one solution to each exercise. Please clearly mark/scratch the solutions that should not be graded.
9. **All solutions must be understandable and well-founded. Write down the important thoughts in clear sentences and keywords. Unless stated otherwise, no points will be awarded for unfounded or incomprehensible solutions.**
10. You may use anything that has been introduced and proved in the lecture or in the exercise sessions. You do not need to re-prove it, but you need to state it clearly and concretely. However, if you need something *different* than what we have shown, you must write a new proof or at least list all necessary changes.
11. Write your student-ID (**Legi-number**) on **all** sheets (and your name only on this cover sheet).

	achieved points (maximum)	reviewer's signature
1	(10)	
2	(8)	
3	(10)	
4	(12)	
5	(10)	
Σ	(50)	

Exercise 1: Short Questions (10 points)

No justification is required.
(There are no negative points for wrong answers.)

(a) **(1+1 points)** Consider the algorithm that given a graph selects a random edge and contracts it until the graph contains only two vertices and finally, it returns the number of edges between these two vertices.

- What is the probability that the algorithm returns the correct min-cut when the input graph is a tree on n vertices?

  Answer: ....

- What is the probability that the algorithm returns the correct min-cut when the input graph is a complete graph on n vertices?

  Answer: ....

(b) **(2 points)** What is the expected and worst-case time needed to find the element of rank k in an array with n elements using the algorithm Quickselect? Express the answer using the $O(\cdot)$ notation.

Answer: ....

(c) **(1+1 points)** What is the expected height of a treap when the keys $\{1, 2, \dots, n\}$ are inserted ...

- in the order $1, 2, 3, \dots, n$, with priorities i.i.d. uniformly random in $[0, 1]$?

  Answer: ....

- in a uniformly random order among the $n!$ permutations, with priorities $1/n, 2/n, \dots, 1$ respectively (i.e. key i has priority i/n)?

  Answer: ....

(d) **(2 points)** Suppose you are given a set of n lines in the plane and you are allowed to use $O(n^2)$ space to store them. What is the query time of an efficient way to determine the number of lines above any given query point?

Answer: ....

....

(e) **(2 points)** Consider a linear program with n variables, $n+1$ inequality constraints, and no equality constraints. How many basic feasible solutions can the linear program have at most?

Answer: ....

....

Exercise 2: Edge contraction and heavy edges (8 points)

Consider the following graph G and let T be the minimum spanning tree of G.

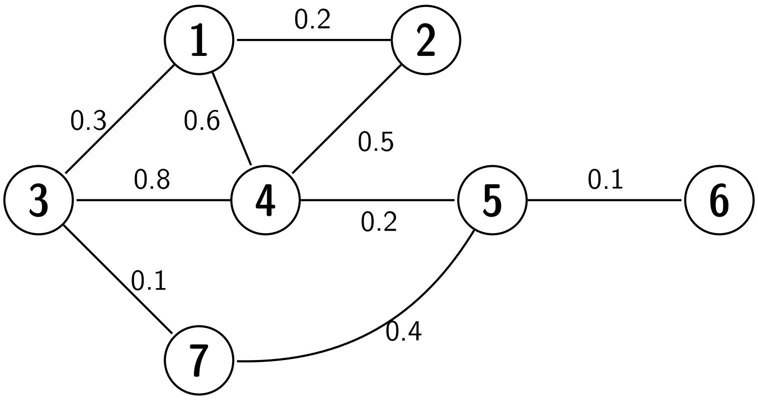

Figure 1: The graph G.

(a) Draw the graph obtained after one iteration of Borůvka's algorithm starting from the graph G. I.e. show the state of the graph after the contractions performed by the algorithm in one call, but before using recursion to compute the MST.

(b) Find the edges in G that are T-heavy.

Exercise 3: Random Binary Search Tree (10 points)

Consider a random search tree with n nodes. What is the expected number of nodes whose left subtree contains exactly three nodes?

Exercise 4: Point Location (12 points)

(a) Let P be a collection of n points in the plane and suppose that for every triple of points in P, the points are not collinear. Consider the set of all the convex quadrilaterals whose vertices are in P and suppose that they all have a different area. Denote with $A(P)$ the area of the quadrilateral with the biggest area and let $A(P) = 0$ if $n < 4$.
We add the points of P one by one in a uniformly random order, yielding a sequence of sets $P_0, P_1, \ldots, P_n$ where $P_0 = \emptyset$ and $P_n = P$.
Compute exactly the expected number of maximal quadrilateral areas changes seen during this process. In other words, compute $\mathbf{E}[|\{i : A(P_i) \neq A(P_{i-1}), 1 \leq i \leq n\}|]$.

Describe the data structures that can solve the following problems. You do not have to analyse the space requirement or the preprocessing time but both have to be polynomially bounded in n.

(b) You are given a collection of n not necessarily distinct lines. For a given query consisting of a line, return the number of lines that intersect with the query line in time $O(\log n)$.

(c) You are given a collection of n rays defined by a line, an origin point and a direction. A query consist of a horizontal line, return all the rays that intersect the line in time $O(k + \log n)$, where k is the number of rays in the solution.

(d) You are given a collection of n rays defined by a line, an origin point and a direction. A query consist of a line, return all the rays that intersect the line in time $O(k + \log n)$, where k is the number of rays in the solution.

Exercise 5: Linear Programming (10 points)

Given a set of n non-vertical lines (described as $a_i x + b_i y = c_i$ for $i = 1, \ldots, n$), write a linear program that finds a point p which minimizes the sum of the distances between p and each line. Write the solution in the form

$$\min c^\top v$$

subject to

$$Av = b$$

and

$$v \geq 0.$$

Hint: The distance between a point (p_x, p_y) *and a line* $a_i x + b_i y = c_i$ *is* $\frac{|a_i p_x + b_i p_y - c_i|}{\sqrt{a_i^2 + b_i^2}}$.