Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Bernd Gärtner, Mohsen Ghaffari, Rasmus Kyng, Angelika Steger, David Steurer

# Algorithms, Probability, and Computing — Midterm Exam — HS21

## 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 | (12) | |
| 2 | (12) | |
| 3 | (12) | |
| 4 | (12) | |
| 5 | (12) | |
| $\Sigma$ | (60) | |

## Exercise 1: Five Short Questions (12 points)

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

(a) Let $G$ denote an $n$-node simple, connected and unweighted graph ($n \geq 100$). Assume that we repeatedly contract an edge chosen uniformly at random until we have a graph with $\lceil \frac{n}{3} \rceil$ vertices, i.e., we call the procedure $\mathrm{RandomContract}(G, \lceil \frac{n}{3} \rceil)$ from the script. Among the four answers, choose the asymptotically tightest lower bound for the probability that the minimum cut size of $G$ is equal to the minimum cut size of the contracted graph. (2 points)
a) $\Omega\left(\frac{1}{n^2}\right)$ b) $\Omega\left(\frac{1}{n}\right)$ c) $\Omega\left(\frac{1}{n^{1/3}}\right)$ d) 0.0001

Answer: ..................................................................

(b) Consider a random binary search tree with the keys 1 up to $n$. For $i, j \in \{1, 2, \ldots, n\}$, we have defined the indicator variable $A_i^j = [\text{node } j \text{ is ancestor of node } i]$. What is $\mathbb{E}\left[A_i^j\right]$? (2 points)

Answer: $\mathbb{E}\left[A_i^j\right] =$ ..................................................................

(c) Given a point set with $n$ points, what are the storage space and query time for the data structure that uses point-line duality and solves the problem of counting the number of points below a query line? (3 points)

Storage space: ..................................................................

Query time: ..................................................................

(d) Let $S$ be a set containing a finite number of non-crossing segments. For a given ordering $\sigma$ of $S$, we denote with $N_{S,\sigma}$ the number of nodes of the history graph of $S$ with respect to $\sigma$.
We define $\mathcal{S}_n = \{S\colon S \text{ is a set containing } n \text{ non-crossing segments in general position}\}$.
Consider the three functions $f, g, h$ with

(i) $f(n) := \min_{S \in \mathcal{S}_n} \min_{\sigma \text{ is an ordering of } S} N_{S,\sigma}$,

(ii) $g(n) := \min_{S \in \mathcal{S}_n} \max_{\sigma \text{ is an ordering of } S} N_{S,\sigma}$ and

(iii) $h(n) := \max_{S \in \mathcal{S}_n} \max_{\sigma \text{ is an ordering of } S} N_{S,\sigma}$.

Determine the asymptotic behavior of the three functions. (3 points)

Answer: $f(n) = \Theta$ .............. $g(n) = \Theta$ .............. $h(n) = \Theta$ ..............

(e) Let $P$ be a polytope in $\mathbb{R}^{10}$ with volume $vol(P) > 0$. We use the ellipsoid method to find a point contained in $P$. The first ellipsoid $E$ has a volume of $vol(E)$. Give a tight (in the worst case) asymptotic upper bound in terms of $vol(P)$ and $vol(E)$ on the number of iterations the ellipsoid method performs until it finds a point in $P$. (2 points)

Answer: $O$ ................................................................

## Exercise 2: Yet Another MSF Algorithm (12 points)

Consider the following variation of the Minimum Spanning Forest Algorithm in the script.

The two differences compared to the algorithm in the script are highlighted in bold. First, the algorithm performs four instead of three Borůvka iterations at the beginning. Second, each edge is sampled with probability $\frac{1}{13}$ instead of probability $\frac{1}{2}$. Let $n$ and $m$ denote the number of vertices and edges of $G$, respectively.

(a) **(5 points)** The algorithm performs two recursive calls. For $i \in \{1, 2\}$, let $G_i = (V_i, E_i)$ be the input graph of the $i$-th recursive call. Give sufficiently tight upper bounds on the expected value of the random variables $N_i = |V_i|$ and $M_i = |E_i|$ in terms of $n$ and $m$. You don't need to provide a justification.

Answer: $\mathbb{E}[N_1] \leq$ ......... $\mathbb{E}[M_1] \leq$ ......... $\mathbb{E}[N_2] \leq$ ......... $\mathbb{E}[M_2] \leq$ .........

(b) **(7 points)**

Using the results from part $a$), provide a **formal and complete** proof that the expected running time of the Minimum Spanning Forest algorithm above is $O(n + m)$. Assume that for $n \leq 100$, the algorithm directly returns the MSF of $G$ without performing recursive calls. In your proof, you are not allowed to refer to the lecture notes. However, you can use the fact that the four Borůvka steps, the edge sampling and the invocation to FINDHEAVY take $O(n + m)$ time (i.e., everything besides the two recursive calls takes $O(n + m)$ time). In case you haven't solved $a$), you can work with the following fictional upper bounds: For $i \in \{1, 2\}$, $\mathbb{E}[N_i] \leq \frac{n}{3}$ and $\mathbb{E}[M_i] \leq \frac{m}{3}$.

## Exercise 3: Random Binary Search Tree (12 points)

Consider a random binary search tree with $n$ nodes. Compute the expected number of nodes that have a left child but no right child.

## Exercise 4: Point Location: Two At A Time (12 points)

Let $S \subseteq \mathbb{R}^2$ be a set of $2n$ points in the plane. We assume that the set of pairwise distances $\{\text{dist}(p, q) : p, q \in S,\ p \neq q\}$ has exactly $\binom{2n}{2}$ elements.

Let $(p_1, \ldots, p_{2n})$ be a permutation of the points in S, drawn uniformly at random from the set of all permutations of S. Imagine that the points are inserted sequentially into the plane according to the order given by the permutation but each time *two* points are inserted simultaneously. After each insertion of two new points we take note of the longest distance between any two of the points currently present. If the longest distance among $p_1, \ldots, p_{2k}$ is different from the longest distance among $p_1, \ldots, p_{2k}, p_{2k+1}, p_{2k+2}$, then we have a *longest distance change*. Here we do not count the insertion of $p_1$ and $p_2$ as a longest distance change (so the first time we may possibly observe a longest distance change is when inserting both $p_3$ and $p_4$).

Compute (exactly) the expected number of longest distance changes. You don't have to simplify your solution.

## Exercise 5: Linear Programming (12 points)

(a) **(6 points)** Consider the following linear program for a given vector $c \in \mathbb{R}^n$

$$\text{minimize } \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad 0 \leq x_1 \leq x_2 \leq \ldots \leq x_n \leq 1.$$

Compute the optimal value and an optimal solution of the linear program (in terms of $\mathbf{c}$). You don't need to prove that your answer is correct.

(b) **(6 points)** A linear program in *standard form* is a linear program of the form

$$\text{maximize } \mathbf{c}^T\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} \leq \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}.$$

Recall that its dual is

$$\text{minimize } \mathbf{b}^T\mathbf{y} \quad \text{subject to} \quad A^T\mathbf{y} \geq \mathbf{c} \text{ and } \mathbf{y} \geq \mathbf{0}.$$

Let $x^* \in \mathbb{R}^n$ be an optimal solution of the primal LP and $y^* \in \mathbb{R}^m$ be an optimal solution of the dual LP. Let $i \in \{1, \ldots, n\}$ be arbitrary. Show that $x_i^* \neq 0$ implies $(A^T y^*)_i = c_i$. That is, if the $i$-th primal variable is nonzero, then the $i$-th dual constraint is tight. Similarly, let $j \in \{1, \ldots, m\}$ be arbitrary. Show that $y_j^* \neq 0$ implies $(Ax^*)_j = b_j$. That is, if the $j$-th dual variable is nonzero, then the $j$-th primal constraint is tight.
Hint: You might want to use strong duality.