ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Algorithms, Probability, and Computing **Final Exam** **HS19**

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 (20 one-sided pages with 7 exercises).
2. You have **3 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	(15)	
2	(15)	
3	(15)	
4	(15)	
5	(15)	
6	(10)	
7	(15)	
Σ	(100)	

Exercise 1: Randomized Quick-Sort (15 points)

We have learned the relation between a random search tree for n keys and randomized quick-sort for n keys. The expected number of comparisons by randomized quick-sort is $O(n \log n)$. Charge comparisons to the pivots. We are interested in the number of comparisons that are charged to the median. For convenience, assume that n is odd, so that the median is the key with rank $\lceil \frac{n}{2} \rceil$.

Complete the following tasks:

(a) **(5 points)** Each key can be viewed as a node in a random search tree. Which property of a node in the tree does the number of comparisons charged to the associated key correspond to? Justify your answer.

(b) **(5 points)** Express the number of comparisons charged to the median in terms of the ancestor indicator random variables A_i^j where $i, j \in [n]$, i.e.,

$$A_i^j = \begin{cases} 1, & \text{if node } j \text{ is ancestor of node } i \text{, and} \\ 0, & \text{otherwise.} \end{cases}$$

(c) **(5 points)** Calculate the expected number of comparisons charged to the median.

Exercise 2: Smallest Enclosing Parallel Rectangle (15 points)

Given a set of n points in the plane, the goal is to find the smallest axis-parallel rectangle that encloses all the n points. This rectangle is determined by the maximum and the minimum x- and y-coordinates among the n points. For simplicity, assume that the x- and y-coordinates of the n points are all distinct. We are interested in a randomized algorithm that inserts the n points one after one in a random order and maintains the up-to-date smallest enclosing axis-parallel rectangle.

Complete the following tasks:

(a) **(5 points)** Analyze the probability that the i-th inserted point does not lie inside the smallest enclosing axis-parallel rectangle of the first $i-1$ inserted points.

(b) **(5 points)** Analyze the expected number of times that the smallest enclosing axis-parallel rectangle changes during the randomized incremental construction.

(c) **(5 points)** Analyze the total construction time of the above-mentioned algorithm.

Exercise 3: The r-Way Minimum Cut (15 points)

For a graph $G = (V, E)$ with n vertices and m edges, an r-way minimum cut is a smallest set of edges in E whose removal breaks the graph G into r or more connected components. The randomized min-cut algorithm (without bootstrapping) can be modified to compute an r-way minimum cut. Complete the following tasks:

(a) **(3 points)** Explain how to modify the randomized min-cut algorithm.

(b) **(4 points)** Let k be the size of an r-way minimum cut and consider the fact that $\binom{n-2}{r-1}m \leq (m-k)\binom{n}{r-1}$. (You do not need to prove this fact.) Use this fact to prove that the probability that the first contraction will not change the size of an r-way minimum cut is at least $\frac{(n-r+1)(n-r)}{n(n-1)}$.

(c) **(4 points)** Prove that the success probability of the whole algorithm is at least

$$\prod_{i=1}^{r-1} \frac{(r-i+1)(r-i)}{(n-i+1)(n-i)}.$$

(d) **(4 points)** Calculate the number of repetitions of the whole algorithm to obtain an r-way minimum cut with probability at least $1 - \frac{1}{n}$.

Exercise 4: Perfect Matching in Bipartite Graph (15 points)

Given an unweighted bipartite graph $G(A \uplus B, E)$, the goal is to find a perfect matching for G. Assume that there exists at least one perfect matching in G. You will first apply linear programming to find a possibly fractional solution, and then modify the fractional solution into an integral one. Enumerate vertices in A from 1 to $|A|$ and vertices in B from 1 to $|B|$, so that (i, j) represents the edge between i-th vertex in A and j-th vertex in B if this edge exists in E. Then, one can define a variable $x_{i,j}$ such that $x_{i,j} = 1$ if (i, j) is contained in the perfect matching and otherwise, $x_{i,j} = 0$.

Complete the following tasks:

(a) **(5 points)** Design a linear program whose integral solutions are perfect matchings in G.

(b) **(5 points)** Prove that a solution from (a) is fractional if and only if there exists an even cycle whose all edges (i, j) are assigned a fractional value to $x_{i,j}$.

(c) **(5 points)** Use the fact from (b) to turn a fractional solution from (a) into an integral one.

(*Hint*: Modify $x_{i,j}$ along the edges (i, j) of a cycle.)

Exercise 5: Randomized Algebraic Algorithms (15 points)

(a) **(5 points)** Let $a \in \{0, 1, 2, \dots, k-1\}^n$ be a vector, and let $r \in_{u.a.r} \{0, 1, 2, \dots, m-1\}^n$ be a random vector. Assume that $k \leq m$, and both k and m are prime numbers. Compute $\Pr[a^T r = 0 \bmod m]$.

(b) **(5 points)** Let A be an $n \times n$ matrix with 0/1-entries. For $1 \leq i, j \leq n$ let $\epsilon_{i,j}$ be independent random variables, $\epsilon_{i,j} \in_{u.a.r.} \{1, 3\}$. Let B be the random matrix with $b_{i,j} = \epsilon_{i,j} \cdot a_{i,j}$. Prove that $\mathbb{E}[\det B] = 2^n \cdot \det A$.

(c) **(5 points)** Let $G = (V_G, E_G)$, $V_G = \{1..n\}$, be a tree of n vertices. Prove that for any orientation $\vec{G}$ of G, $\det\left(A_S(\vec{G})\right) \leq 1$. Recall that

$$\begin{aligned} A_s(\vec{G}) &= (a_{ij})_{i,j=1}^n \in \{0, +1, -1\}^{n \times n} \text{ , where} \\ a_{ij} &:= \begin{cases} +1 & \text{if } (i,j) \in E_{\vec{G}}, \\ -1 & \text{if } (j,i) \in E_{\vec{G}}, \text{ and} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Hint: Prove that there exists at most one perfect matching in a tree.

Exercise 6: Parallel Sampling (10 points)

Design a randomized parallel algorithm that samples exactly k elements uniformly at random from the n input elements without replacement. The required expected depth is $\mathcal{O}(\log n)$ and the required expected total work is $\mathcal{O}(n \log n)$.

Exercise 7: Parallel Heavy-Light Decomposition (15 points)

Consider a rooted tree T with n vertices and a root r. For each vertex v in T, its neighbors are stored in a linked list L_v. Moreover, let T_v denote the subtree of T rooted at v and let $p(v)$ denote the parent of v. Complete the following two tasks:

(a) **(8 points)** A vertex u is a *light* child if $|T_u| < \frac{1}{2}|T_{p(u)}|$, i.e., the size of the subtree rooted at u is less than half the size of the subtree rooted at $p(u)$. Design a parallel algorithm to identify all light children with $\mathcal{O}(\log n)$ depth and with $\mathcal{O}(n)$ total work.

(b) **(7 points)** The *light root* of a vertex v is defined recursively as follows:

$$lr(v) = \begin{cases} lr(p(v)) & \text{if both } v \text{ and } p(v) \text{ are light children,} \\ v & \text{if } v \text{ is a light child but } p(v) \text{ is not,} \\ \emptyset & \text{otherwise.} \end{cases}$$

One can imagine to recursively trace the parents until the last one which is not a light child of its parent. Design a parallel algorithm to compute the light root for all the vertices with $\mathcal{O}(\log \log n)$ depth and with $\mathcal{O}(n \log \log n)$ total work.

Hint: Think about the largest possible length of the path between a vertex and its light root.