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 — Final Exam — HS21

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: ....

Instructions

1. The exam consists of 5 exercises.
2. Check your exam documents for completeness (18 one-sided pages with 5 exercises).
3. You have **3 hours** to solve the exercises.
4. 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.
5. If you're unable to take the exam under regular conditions, immediately inform an assistant.
6. **Pencils** are not allowed. Pencil-written solutions will not be reviewed.
7. No auxiliary material is allowed. Electronic devices must be turned off and should not be on your desk. We will write the current time on the blackboard every 15 minutes.
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 proven 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	(20)	
2	(20)	
3	(20)	
4	(15)	
5	(25)	
Σ	(100)	

Exercise 1: Random Binary Search Tree (20 points)

Consider a random binary search tree on the nodes $\{1, 2, \ldots, n\}$. We denote by X_n the number of nodes whose left or right subtree (or both) have exactly one node. We define $x_n = \mathbb{E}[X_n]$.

(a) **(4 points)** Compute x_0, x_1, x_2 and x_3.

(b) **(8 points)** Find a recurrence formula for x_n, for all $n \geq 4$.

(c) **(8 points)** Solve the above recurrence formula for all $n \geq 4$.

Exercise 2: Backwards Analysis (20 points)

Let π be a permutation of $\{1, 2, \ldots, n\}$ chosen uniformly at random. For $i \in \{1, 2, \ldots, n\}$, we denote by A_i the event that $\pi(i) = \min_{j \in \{1,2,\ldots,i\}} \pi(j)$.

(a) **(7 points)** Show that $\Pr[A_i] = \frac{1}{i}$ for $i \in \{1, 2, \ldots, n\}$.

(b) **(8 points)** Show that the events $A_1, A_2, \ldots, A_n$ are mutually independent. That is, for every $k \in \{2, 3, \ldots, n\}$ and indices $1 \leq i_1 < i_2 < \ldots < i_k \leq n$, it holds that $\Pr[A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k}] = \Pr[A_{i_1}] \cdot \Pr[A_{i_2}] \cdot \ldots \cdot \Pr[A_{i_k}]$.

(c) **(5 points)** For $i \in \{1, 2, \ldots, n\}$, let X_i be the indicator variable for the event A_i and $X = \sum_{i=1}^{n} X_i$. Show that $\Pr[X \geq 100(\ln(n) + 1)] \leq n^{-5}$.
Hint: You can use without proof that $\ln(n+1) \leq \sum_{i=1}^{n} \frac{1}{i} \leq \ln(n) + 1$ and the following concentration bound: Let $X_1, X_2, \ldots, X_n$ be mutually independent and Bernoulli-distributed random variables, $X := \sum_{i=1}^{n} X_i$ and $\delta \geq 0$. Then,
$\Pr[X \geq (1+\delta)\mathbb{E}[X]] \leq e^{-\frac{1}{3}\min(\delta, \delta^2)\mathbb{E}[X]}$.

Exercise 3: Linear Programming (20 points)

Let $G = (V, E)$ be a graph. Consider the following linear program with one variable x_e for each edge $e \in E$

$$\begin{array}{lll} \text{maximize} & \sum_{e \in E} x_e & \\ \text{subject to} & \sum_{e \in E: v \in e} x_e \leq b_v & \text{for every } v \in V \\ & x_e \leq r_e & \text{for every } e \in E \\ & \mathbf{x} \geq \mathbf{0}, & \end{array}$$

with $\mathbf{b} \in \mathbb{R}^V_{\geq 0}$ and $\mathbf{r} \in \mathbb{R}^E_{\geq 0}$ being two nonnegative vectors. We denote with OPT the optimum value of the linear program.

(a) **(6 points)** Write down the dual of the linear program.

(b) **(7 points)** Let $\mathbf{x}$ be a feasible solution of the linear program such that for every $e = \{u, v\} \in E$ at least one of the following three things holds:

- $x_e = r_e$
- $\sum_{e \in E:\, v \in e} x_e = b_v$
- $\sum_{e \in E:\, u \in e} x_e = b_u$

Show that $\sum_{e \in E} x_e \geq \frac{1}{3}\mathrm{OPT}$.
Remark: This exercise builds upon exercise a). However, even if you have not solved a), you can still obtain partial points by providing a high-level explanation how the dual could in principle be used to derive the desired lower bound.

(c) **(7 points)** Devise an algorithm with running time $O(|V| + |E|)$ that finds a feasible solution $\mathbf{x}$ of the linear program with $\sum_{e \in E} x_e \geq \frac{1}{3}\mathrm{OPT}$.

Exercise 4: Randomized Algebraic Algorithms (15 points)

Let $G = (V, E)$ be a bipartite graph with $2n$ nodes and bipartition $V = \{u_1, u_2, \ldots, u_n\} \sqcup \{v_1, v_2, \ldots, v_n\}$. For each edge $\{u_i, v_j\} \in E$, we introduce one variable x_{ij}. In the lecture, we have defined an $n \times n$ matrix A by setting

$$a_{ij} = \begin{cases} x_{ij} & \text{if } \{u_i, v_j\} \in E, \\ 0 & \text{otherwise} \end{cases}$$

for $i, j \in \{1, 2, \ldots, n\}$.

Recall that the rank $\mathrm{rk}(M)$ of a matrix M is the maximum number of linearly independent rows/columns of M. Note that A is not a standard matrix in the sense that some entries are variables. Let S_A denote the set consisting of all matrices that one can obtain from A by fixing the variables in A in an arbitrary way. We define the rank of A as $\mathrm{rk}(A) = \max_{M \in S_A} \mathrm{rk}(M)$. Let k denote the size of the largest matching in G.

(a) **(7 points)** Show that $\mathrm{rk}(A) \geq k$.

(b) **(8 points)** Show that $\mathrm{rk}(A) \leq k$.

Exercise 5: Parallel Algorithms (25 points)

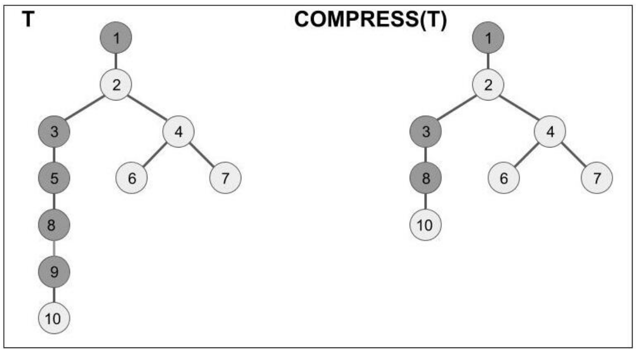

Figure 1: Every node with exactly one child is colored in dark grey. Each connected component of dark grey nodes corresponds to a maximal highway. The two maximal highways in T are 1 and $3, 5, 8, 9$. Shortening the highway $3, 5, 8, 9$ results in the highway $3, 8$.

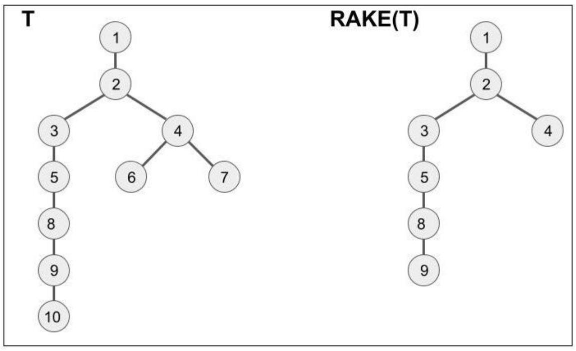

Figure 2: One obtains the rooted tree RAKE(T) by removing the leaves $6, 7$ and 10 from T.

Let T be an n-node rooted tree. A sequence of nodes $v_1, v_2, \ldots, v_k$ is a highway if every node has *exactly* one child and for $i \in \{1, 2, \ldots, k-1\}$ the child of v_i is v_{i+1}. We say that $v_1, v_2, \ldots, v_k$ is a maximal highway if there does not exist a node u such that either $u, v_1, v_2, \ldots, v_k$ or $v_1, v_2, \ldots, v_k, u$ is a highway. A highway is *shortened* by removing every other vertex on it, i.e., by removing $v_2, v_4, v_6, \ldots$, and turning it into a highway that connects v_1 to v_3, v_3 to v_5 and so on. Also, if v_k is removed, then v_{k-1} will be connected to the child of v_k .

The COMPRESS operation applied to T simultaneously shortens every maximal highway of T. The RAKE operation applied to T removes all the leaves of T (assuming $n \geq 2$). Both operations are illustrated in Figure 1 and Figure 2, respectively.

For all the exercises below, we only consider rooted trees with each node having at most 2 children. Such a tree is stored by every node storing a pointer to its parent and a pointer to each child.
The computational model we use is CRCW PRAM and all of the algorithms have to be deterministic.

(a) **(3 points)** Show how to implement the RAKE operation in $O(n)$ work and $O(1)$ depth.

(b) **(4 points)** Show how to implement the COMPRESS operation in $O(n \log n)$ work and $O(\log n)$ depth.

(c) **(5 points)** Assume T has at least $0.9n$ nodes with exactly one child. Show that applying the COMPRESS operation to T results in a rooted tree with at most $c \cdot n$ nodes for some fixed constant $c < 1$.

(d) **(5 points)** Assume T has $n \geq 2$ nodes. Let $T' = \text{COMPRESS}(T)$ and $T'' = \text{RAKE}(T')$. Show that T'' has at most $c' \cdot n$ nodes for some fixed constant $c' < 1$.

(e) **(8 points)**

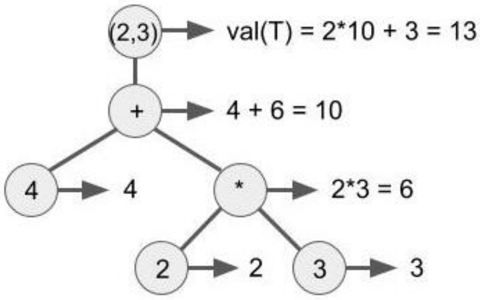

Figure 3: The figure depicts an expression tree and the evaluation of each subtree.

An expression tree is a rooted tree with four different types of nodes.

- Each leaf node is equipped with a number x.
- Each node with exactly one child is equipped with a pair of numbers (a, b).
- Each node with two children is either a "$+$"-node or a "$*$"-node.

Each expression tree T evaluates to a number $val(T)$. If T consists of just a single node equipped with the number x, then $val(T) = x$. If the root node of T is associated with a pair (a, b), then $val(T) = a \cdot val(T') + b$ where T' is the subtree of the only child of the root. If the root node of T is an "OP"-node for $OP \in \{+, *\}$, then $val(T) = val(T_1)$ OP $val(T_2)$ with T_1 and T_2 being the subtrees of the two children of the root. Figure 3 depicts an expression tree together with its evaluation.

Show how to evaluate an expression tree in $O(n \log n)$ work and $O(\log^2 n)$ depth. Hint: You can obtain partial points if your algorithm needs $O(n \log^2 n)$ work.