ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Algorithms, Probability, and Computing | **Special Assignment 2** | **HS20**

- Write an exposition of your solution using a computer, where we strongly recommend to use LaTeX. **We do not grade hand-written solutions.**
- You need to submit your solution via Moodle until **Tuesday, December 1, 2020** by **10 pm**. Late solutions will not be graded.
- For geometric drawings that can easily be integrated into LaTeX documents, we recommend the drawing editor IPE, retrievable at `http://ipe.otfried.org` in source code and as an executable for Windows.
- Write short, simple, and precise sentences.
- This is a theory course, which means: if an exercise does not explicitly say "you do not need to prove your answer" or "justify intuitively", then a formal proof is **always** required. You can of course refer in your solutions to the lecture notes and to the exercises, if a result you need has already been proved there.
- We would like to stress that the ETH Disciplinary Code applies to this special assignment as it constitutes part of your final grade. The only exception we make to the Code is that we encourage you to verbally discuss the tasks with your colleagues. However, you need to write down the names of all your collaborators at the beginning of the writeup. It is strictly prohibited to share any (hand)written or electronic (partial) solutions with any of your colleagues. We are obligated to inform the Rector of any violations of the Code.
- There will be two special assignments this semester. Both of them will be graded and the average grade will contribute 20% to your final grade.
- As with all exercises, the material of the special assignments is relevant for the (midterm and final) exams.

Exercise 1 30 points

(*Geometry of Polytopes*)

Let $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a polyhedron with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ and let $y \in P$ be a point in the polyhedron. Prove that the following three statements are equivalent.

(i) The point y is a basic feasible solution of P, i.e., y satisfies n linearly independent constraints with equality.

(ii) There do not exist two distinct points $y_1, y_2 \in P$ with $y = \frac{1}{2}(y_1 + y_2)$.

(iii) The point y is the unique maximizer of P for some linear function, i.e., there exists a $c \in \mathbb{R}^n$ such that for every $z \in P \setminus \{y\}$, $c^T y > c^T z$.

Exercise 2 30 points

(*Vertex Cover on Bipartite Graphs*)

In this exercise, we consider the minimum vertex cover problem on a bipartite graph $G = (V, E)$. Note that the two sides of the bipartition don't need to necessarily have the same size. A vertex cover is a subset of the vertices $V_{cover} \subseteq V$ with the property that at least one endpoint of each edge is contained in V_{cover}. The goal is to find a vertex cover of minimum size. We will use linear programming to devise an algorithm that solves this problem in polynomial time.

(a) Devise an integer linear program with the following property: Each feasible solution to the integer linear program with objective value z corresponds to a vertex cover with z vertices and vice versa.

(b) Now, relax the integer linear program. Then, devise a procedure that given a feasible fractional solution to the relaxed linear program with objective value z, it finds a vertex cover of size at most z in polynomial time.
Note: This shows that the integrality gap of the linear program is 1.

(c) Use the previous exercises to devise an algorithm that solves the minimum vertex cover problem on bipartite graphs in polynomial time.

Exercise 3 40 points

(*Isomorphism of Rooted Trees*)

You might have encountered the graph isomorphism problem during your studies. Given two graphs G_1 and G_2, it asks — informally speaking — whether it is possible to relabel the vertices of G_1 such that G_1 and G_2 are the same graph. On the one hand, the problem is not known to be NP-complete, on the other hand, no polynomial time algorithm is known. Here, we consider a special case of the problem. The input consists of two n-node rooted trees T_1 and T_2. Now, the question is to decide whether the two rooted trees are isomorphic. We refer to two rooted trees as isomorphic if and only if there exists a bijective function f from the vertices of T_1 to the vertices of T_2 such that for each vertex v of T_1 with children $v_1, v_2, \ldots, v_k$, the children of $f(v)$ in T_2 are exactly $f(v_1), f(v_2), \ldots, f(v_k)$. We will devise a randomized algorithm that solves this decision problem in linear time. To that end, for a given rooted tree T, we associate a polynomial P_v with each node v in T. The polynomials are defined recursively. The base case is that for each leaf vertex u, we define $P_u := x_h$, where h is the height of u in the tree. Now, let v be an internal vertex of height h in the tree, and let $v_1, v_2, \ldots, v_k$ be its children. We assign the polynomial

$$P_v := (x_h - P_{v_1}) \cdot (x_h - P_{v_2}) \cdot \ldots \cdot (x_h - P_{v_k})$$

with the node. Note that we have exactly one variable for each level of the tree.
Remark: You can solve exercises b) ,c) and d) without solving exercise a).

(a) Let T_1 and T_2 denote two rooted trees with roots r_1 and r_2, respectively. Show that T_1 and T_2 are isomorphic if and only if P_{r_1} and P_{r_2} are the same polynomial, where we assume that the underlying field of the polynomials is $\mathbb{R}$.

(b) Let $S = \{1, 2, \ldots, n^2\}$ and let h denote the height of T_1 and T_2 (WLOG, we assume that T_1 and T_2 have the same height). Let $x = (x_0, x_1, \ldots, x_h)$ be chosen uniformly at random from S^{h+1}. Show that if P_{r_1} and P_{r_2} are not the same polynomial, then $P_{r_1}(x) \neq P_{r_2}(x)$ with probability $1 - O(1/n)$.

(c) Let p be a prime chosen uniformly at random from the set of all primes in the range from 1 to n^3 (We assume that $n \geq 2$). Let $x \in S^{h+1}$ such that $P_{r_1}(x) \neq P_{r_2}(x)$. Show that $P_{r_1}(x) \mod p \neq P_{r_2}(x) \mod p$ with probability $1 - O(1/n)$.
Hint 1: Let N denote some integer. Observe that the number of distinct prime divisors of N is $O(\log |N|)$.
Hint 2: For each $k \in \mathbb{N}$, let $h(k)$ denote the number of primes less than k. You can use without a proof that $h(k) = \Omega(k/\log(k))$.

(d) By using the previous results, devise a randomized algorithm that takes as input two rooted trees. If the two rooted trees are isomorphic, the algorithm should always output "Yes". If the two rooted trees are not isomorphic, the algorithm should output "No" with probability at least $1 - O(1/n)$. Your algorithm is only allowed to operate on integers that can be represented with $O(\log(n))$-bits, i.e., integers between $-n^c$ and n^c for some constant c that you are free to choose. All standard arithmetic, logic and control flow operations are assumed to have unit cost. Furthermore, you can assume that picking a number uniformly at random from the set $\{1, 2, \ldots, N\}$ for some $N \leq n^c$ also has a unit cost. Your algorithm should always run in $O(n)$ time.

Hint: For a given natural number k, you can use without a proof that one can deterministically decide in $O(\log^{100} k)$ time whether k is a prime number or not.