HW 10

Due on **Wednesday, November 29, 2023 at 9pm**

CS 473: Algorithms, Fall 2023 Version: **1.2**

Submission instructions: As in previous homeworks.
Read carefully the class notes before answering the following questions.
https://courses.engr.illinois.edu/cs473/fa2023/lec/notes/25_lp.pdf
https://courses.engr.illinois.edu/cs473/fa2023/lec/notes/26_lp_II.pdf
https://courses.engr.illinois.edu/cs473/fa2023/lec/notes/27_lp_III.pdf

28 (100 PTS.) Linear programming.

Let L be a linear program given in slack form, with n nonbasic variables N, and m basic variables B. Let N' and B' be a different partition of $N \cup B$, such that $|N' \cup B'| = |N \cup B|$. Show a polynomial time algorithm that computes an equivalent slack form that has N' as the nonbasic variables, and B' as the basic variables. How fast is your algorithm? (Note, that by definition, it must be that $|B| = |B'|$, and $|N| = |N'|$.)

29 (100 PTS.) Linear programming II. Provide *detailed* solutions for the following problems, showing each pivoting stage separately.

29.A. (25 PTS.)
maximize $6x_1 + 8x_2 + 5x_3 + 9x_4$
subject to
$2x_1 + x_2 + x_3 + 3x_4 \leq 7$
$x_1 + 3x_2 + x_3 + 2x_4 \leq 8$
$x_1, x_2, x_3, x_4 \leq 10.$

29.B. (25 PTS.)
maximize $2x_1 + 2x_2$
subject to
$2x_1 + x_2 \leq 14$
$2x_1 + 3x_2 \leq 5$
$4x_1 + x_2 \leq 5$
$x_1 + 5x_2 \leq 3$
$x_1, x_2 \geq 0.$

29.C. (25 PTS.)
maximize $6x_1 + 8x_2 + 4x_3 + 9x_4$
subject to
$x_1 + x_2 + x_3 + x_4 = 2$
$x_1, x_2, x_3, x_4 \geq 0.$

29.D. (25 PTS.)
minimize $x_{12} + 8x_{13} + 9x_{14} + 2x_{23} + 7x_{24} + 3x_{34}$
subject to
$x_{12} + x_{13} + x_{14} \geq 1$
$-x_{12} + x_{23} + x_{24} = 0$
$-x_{13} - x_{23} + x_{34} = 0$
$x_{14} + x_{24} + x_{34} \leq 2$
$x_{12}, x_{13}, \ldots, x_{34} \geq 0.$

30 (100 PTS.) Triangles and points.

You are given a set P of n points in the plane, and a set T of n triangles. Two triangles $\triangle, \triangle'$ are in ***conflict*** if there is a point $p \in P$, such that $p \in \triangle \cap \triangle'$. Our task at hand is to compute the largest conflict-free subset $S \subseteq T$ of triangles.

30.A. (20 PTS.) Write an LP that (fractionally) solves this problem. For simplicity, from this point on, we assume the value of the optimal fractional solution, denoted by α^*, is at least (say) $20n^{2/3}$.

30.B. (20 PTS.) Consider picking triangles from T according to the fractional values of the optimal solution of the LP. Using the following theorem prove that in expectation, the expected number of triangles chosen is α^*, and the number of conflicting pairs of triangles in the sample is $O(n)$. (Hint: Prove that each point of P, in expectation, give rise to a constant number of conflicting pairs.)

> **Theorem 10.1 (Chernoff's inequality).** *Let $x_1, x_2, \ldots, x_n \in [0,1]$ be n numbers, such that $\sum_{i=1}^{n} x_i \leq 1$. For $i = 1, \ldots, n$, let $X_i \in \{0,1\}$ be a random variable that is picked to be 1 (independently) with probability x_i. Then, for any $t \geq 5$, we have $\mathbb{P}[\sum_{i=1}^{n} X_i \geq t] \leq 2^{-t}$.*

30.C. (60 PTS.) Describe an efficient algorithm (assuming that LP can be solved efficiently) that outputs, in expectation, a subset $S \subseteq T$ of size $\Omega(n^{1/3})$, such that S is conflict free. Prove the bound on the size of S.

31 (100 PTS.) Color or not.

Let $G = (V, E)$ be an undirected graph over n vertices. Assume that you are given a numbering $\pi : V \to \{1, \ldots, n\}$ (i.e., every vertex have a unique number), such that for any edge $ij \in E$, we have $|\pi(i) - \pi(j)| \leq 30$.

Either prove that it is NP-HARD to compute a 3-coloring of G, or provide a polynomial time algorithm.

32 (100 PTS.) Beware of Greeks bearing gifts

The deduction faun, came to visit you on labor day, and left you with three black boxes.

32.A. (30 PTS.) The first black-box solves Partition in polynomial time (note that this black box solves the decision problem). Let S be a given set of n integer numbers. Describe a polynomial time algorithm that computes, using the black box, a partition of S if such a solution exists. Namely, your algorithm should output a subset $T \subseteq S$, such that $\sum_{s \in T} s = \sum_{s \in S \setminus T} s$.

32.B. (30 PTS.) The first black box can solve the following decision problem in polynomial time:

Minimum Distinguisher

> **Instance**: A finite set I of "possible illnesses," a collection $\mathcal{F}$ of subsets of I, representing "tests," and a positive integer α.
> **Question:** Is there a subcollection $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \leq \alpha$, such that, for every triple of (distinct) illnesses $i_i, i_j, i_k \in I$, there is some test $c \in \mathcal{F}'$ for which $1 \leq |\{i_i, i_j, i_k\} \cap c| \leq 2$ (that is, a test c that "distinguishes" between the three illnesses?)

Using this black box, solve, in polynomial time, the optimization version of this problem (i.e., finding and outputting the smallest possible set $\mathcal{F}'$).

32.C. (40 PTS.) The second box solves the following problem.

Subgraph Isomorphism

Instance: Two graphs, $G = (V_1, E_1)$ and $H = (V_2, E_2)$.
Question: Does G contain a subgraph *isomorphic* to H, that is, a subset $V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$, $|e| = |E_2|$, and there exists a one-to-one function $f : V_2 \to V$ satisfying $\{u, v\} \in E_2$ if and only if $\{f(u), f(v)\} \in E$?

Show how to use this black box, to compute the subgraph isomorphism (i.e., you are given G and H, and you have to output f) in polynomial time.

33 (100 PTS.) Polytime reduction

33.A. (50 PTS.) Provide a polynomial time reduction from Partition to Origin.

Origin

Instance: A set $V = \{v_1, \ldots, v_n\}$ of n integer vectors in the plane (i.e., the coordinates of the v_is are integer numbers.)
Question: Is there a choice of n numbers $\alpha_i \in \{-1, +1\}$ such that $\sum_i v_i \alpha_i = (0, 0)$?

33.B. (50 PTS.) (Harder?) Provide a polynomial time reduction from Origin to Partition.

Both reductions are allowed to be randomized, but then you have to show that the probability they work is arbitrarily close to one. You can also safely assume here that basic operations on integers take $O(1)$ time, even if the integer numbers are large.