Homework #4:

1. input: r, maximization LP
   is $Opt(LP) \geq r$?
   → show that belongs to both NP and CoNP
   given the primal: $\max c^T \cdot \vec{x}$ s.t. $A\vec{x} \leq \vec{b}$, $\vec{x} \geq 0$ → we have the dual: $\min b^T \cdot \vec{y}$ s.t. $A^T y \geq c$, $y \geq 0$

   Must prove: this $LP \in NP \cap CoNP$
   1st prove $LP \in NP$: efficient certifier for LP:
   - Take (LP, r) input
   - See if $Opt(LP) \geq r$
     If yes output yes
     If no output no

   ... can this be verified efficiently?
   → in order to verify $Opt(LP) \geq r$
   (can NOT use the ellipsoid method $\Rightarrow LP \in P$) this is our certifier
   if LP is feasible and bounded then let $\vec{x}^*$ be a vertex s.t. $\vec{x}^*$ satisfies all constraints AND s.t. $c^T\vec{x}^* \geq r$... it takes polynomial time to check that $A\vec{x}^* \leq \vec{b}$ and $\vec{x}^* \geq 0$ and $c^T\vec{x}^* \geq r$
   This sufficiently proves $LP \in NP$ ← switches yes inputs of LP to no and no inputs of LP to yes
   2nd prove $LP \in Co\text{-}NP \Rightarrow \overline{LP} \in NP$ where $\overline{LP}$ is defined as:
   Input: $\max c^T\vec{x}$ s.t. $A\vec{x} \leq c$, $\vec{x} \geq 0$ → Q: Is $Opt(\overline{LP}) < r$

   we can use strong duality to show that

   $\max c^T\vec{x}$ s.t. $A\vec{x} \leq c$, $\vec{x} \geq 0$ — $Opt(Primal) = Opt(Dual)$ — $\min b^T \cdot \vec{y}$ s.t. $A^T y \leq c$, $y \geq 0$

   $\Rightarrow Opt(Dual) < r \Rightarrow Opt(\overline{LP}) < r$

   This proves $LP \in co\text{-}NP$
   $\therefore LP \in NP \cap co\text{-}NP$

2. coordinates of n points $p_1, \ldots, p_n$ of plane
→ find non-overlapping disks centered on these points s.t. the sum of the radii is maximized

(a) variables: $r_i, d(p_i, p_j)$ $\forall i,j$ (where r_i is the radius of of p_i's disk and $d(p_i,p_j)$ is the distance between points p_i and p_j)

$$\max: \sum_{i=1}^{n} r_i$$
$$w_{ij} \times (r_i + r_j \leq d(p_i, p_j)) \quad \forall i,j$$
$$r_i \geq 0 \quad \forall i$$

(b) directed graph on points $p_1, \ldots, p_n$ where for every i and j, there is an edge from p_i to p_j and one edge p_j to p_i both with a cost equal to $d(p_i, p_j)$
→ we want to cover vertices of graph with cycles (each vertex belongs to at least one cycle) so that the total sum of the cost of the edges that participate in the cycles is minimized)

Dual of (a): minimize $\sum w_{ij} d(p_i, p_j)$ — vars: $w_{ij}, d(p_i,p_j)$ $\forall i,j$ (where w_{ij} is the weight for a directed edge between p_i and p_j)

$$\text{s.t.} \quad \sum_j w_{ij} \geq 1 \quad \forall i$$
$$w_{ij} \geq 0 \quad \forall i,j$$

LP duality states that the Dual provides an upper bound for the primal
→ the dual here is trying to minimize the sum of the weights of directed edge-ij * distance from p_i to p_j
note that the primal is attempting to maximize the radii of all the disks where as the dual is attempting to minimize the weight of edge from p_i to p_j for all p_i and p_j ... the radius of p_i's disk + the radius of p_j's disk must be $\leq$ the weight of the edge from p_i to p_j
since we are trying to find the minimum cycle cover that (let's say) uses w_{ij} and w_{ji}, and we know that $w_{ij} + w_{ji} \geq 2 * (r_i + r_j)$
we can conclude that the solution to this problem is at least twice the solution to the non-overlapping disk problem

4. input: n numbers (each in decimal representation), # K

→ can we select K of these numbers so that if we add them up, the digit 2 will <u>NOT</u> appear in the decimal representation of the sum

→ prove (let's call this problem) SUMS is NP-complete!

<u>proof</u>: 1^{st} show SUMS ∈ NP ← *note: N = set of n numbers

Certifier: Takes (N, K) and S where S is a subset of the numbers of N

→ we output <u>yes</u> if $|S| = k$ <u>and</u> in the decimal representation of the sum of all the numbers in S, there does NOT exist the digit 2

→ else, we output <u>no</u>

2^{nd} we must show completeness by proving SAT $\leq_p$ SUMS (because SAT is a known NP-complete)

→ we want to solve SAT using an oracle for SUMS

let us represent our numbers by a matrix where there are n rows and the entry of each column corresponds to each individual digit of the nth number (making sure to line the numbers up accordingly so the 1^{st} column will represent the highest digit place of all the numbers) ← so pad the numbers that don't have digits here with 0's

Ex: N = {102.35, 36, 7.002, 321.567}

$$\begin{array}{cccccc} 1 & \overset{+1}{0} & 2 & \overset{+1}{3} & 5 & 0 \\ 0 & 3 & 5 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 2 \\ 3 & 2 & 1 & 5 & 6 & 7 \\ \hline 4 & 6 & 5 & 9 & 1 & 9 \end{array}$$

\+

sum → 4 6 5 9 1 9 ✓ yes

(make sure to start from left most column so if the sum needs to be carried to the column to the right it can)

...so now proving SAT $\leq_p$ SUMS!

Input: CNF formula $\emptyset$

Ex: $\emptyset = (\overset{F}{x_1} \vee \overset{T}{x_2} \vee \overset{F}{\bar{x}_3}) \wedge (\overset{T}{\bar{x}_1}) \wedge (\overset{F}{\bar{x}_2} \vee \overset{T}{x_3}) \wedge (\overset{T}{x_3})$

$x_1 = F$
$x_3 = T$
$x_2 = T$

(#) let this by the following "sum matrix"

$x_1 = T \Rightarrow t_1 \in S$

$\bar{x}_1 = T \Rightarrow f_1 \in S$

N	x_1	x_2	x_3	C_1	C_2	C_3	C_4
t_1	1	0	0	1	0	0	0
f_1	1	0	0	0	1	0	0
t_2	0	1	0	1	0	0	0
f_2	0	1	0	0	0	1	0
t_3	0	0	1	0	0	1	1
f_3	0	0	1	1	0	0	0
*	0	0	0	2	2	2	2

$\rightarrow$ so $t_1 = 1001000$
⋮

K = # of variables in $\emptyset$

let K=3 $\rightarrow$ <u>must</u> select t_3 and f_1 (or else sum would equal 2) $\rightarrow$ now we have either t_2 or f_2 remaining (because if chose either t_1 or f_3 sum would equal 2) $\rightarrow$ finally must select t_2 to satisfy C_1 column

	x_1	x_2	x_3	C_1	C_2	C_3	C_4
f_1	1	0	0	0	1	0	0
t_3	0	0	1	0	0	1	1
t_2	0	1	0	1	0	0	0
+ *	0	0	0	2	2	2	2
	1	1	1	3	3	3	3 ✓

Oracle Alg:

- Construct chart # from $\emptyset$
  - $\rightarrow$ rows representing t_i and f_i for each variable x_i and columns representing $\forall x_i$ followed by $\forall c_j$
  - $\rightarrow$ for $\forall x_i$ columns, put a 1 in $t_i x_i$-th entry <u>and</u> $f_i x_i$-th entry and 0 for all other entries in the column
  - $\rightarrow$ for $\forall c_j$ columns, put a 1 in t_i-th row if x_i is in C_j, put a 1 in the f_i-th row if $\bar{x}_i$ is in C_j (and 0 for all other entries in the column)
  - $\rightarrow$ finally include a "*" row in which all the $*x_i$-th entries are 0 and all the $*c_j$-th entries are 2

<u>note</u>: each digit of each t_i and f_i is representative of that element's column entries (ie. if t_i's column entry at $x_1 = 1$, $x_2 = 0$, $x_3 = 0$ and so on, decimal rep of t_i will be 100...)

- If (let's call our set of t_i's and f_i's) N of # has a subset S of size K of which the sum of these K elements <u>ALONG WITH</u> the sum of element "*" (so * <u>must</u> be included in the sum) satisfies the constraint that the digit 2 will <u>NOT</u> appear in the decimal representation of the sum
  - $\rightarrow$ Then output yes (SAT is satisfiable w this input)
  - $\rightarrow$ else, output No (SAT is NOT satisfiable w this input) $\Longrightarrow$ SAT $\leq_p$ SUMS

3. input: n x m table → n students, columns: fav things

k → select k so that every two of them share at least 1 fav

prove (let's call this) FAV's is NP-complete!

proof: 1^{st} show FAVS ∈ NP

Certifier: Takes (M, K) and S where S is a subset of the students of matrix M

→ we output Yes if |S| = K and every two students of S share at least 1 fav thing in common

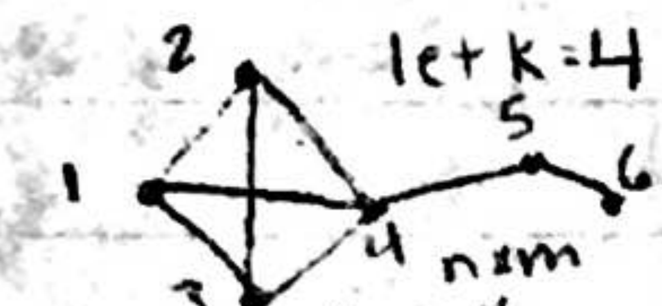
→ else, we output no

2^{nd} show the completeness by proving CLIQUE $\leq_p$ FAVS (because CLIQUE is a known NP-complete)

→ we want to solve CLIQUE using an oracle for FAVS

Input: Graph G, k ∈ ℕ

Q: Does G contain a clique of size k?



let k = 4

n×m

construct an n×m matrix H (where n = # vertices and m = # of edges)

H

X	X	X	m	q	u	z	d
X	e	j	X	X	v	a	c
a	X	i	X	r	X	b	f
b	f	X	n	X	X	X	g
c	g	k	o	s	w	X	X
d	h	l	p	t	y	c	X

S = {1, 2, 3, 4}

K = 4 → is there a set S s.t. every 2 share at least 1 fav

Oracle Alg:

- Construct n×m matrix H where n = # vertices and m = # edges
  → For each column of H (∀C_i of H)
  if ∃ an edge between vertex i and vertex j in G, put an "x" i and j rows of that column, fill in the rest of the row entries of that column with arbitrary entries (could be letters, #'s, words... as long as they are different within the same column)
- If H has a subset of size k in which every two students share at least 1 favorite thing (both have x in the same column)

  Then output Yes (G has a clique of size k)

  Otherwise, output No (G does not have a clique of size k)

⇒ VC $\leq_p$ FAVS

5. Prove NP-Completeness!

Input: CNF ϕ, positive integer k

Q: Does ϕ have a truth assignment that assigns True to <u>half</u> the terms in each clause?

EX: $x_1 = T$, $x_2 = F$, $x_3 = F$, $x_4 = F$ assigns True to exactly half the terms in each clause of $\phi = (x_1 \vee \bar{x}_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2)$

<u>proof</u>: 1st prove (let's call this problem) $\frac{1}{2}$SAT $\in$ NP

certifier: Takes (ϕ, k) and θ where θ is a truth assignment to the variables. If θ satisfies ϕ and θ satisfies the constraint that True must be assigned to exactly half the terms in each clause of ϕ

Then output <u>yes</u>

Otherwise, output <u>NO</u>

2nd show the completeness of the problem by proving SAT $\leq_p$ $\frac{1}{2}$SAT (because SAT is a known NP-complete)

→ we want to solve SAT using an oracle for $\frac{1}{2}$SAT

Input: A CNF ϕ

Q: Is there a truth assignment to variables that satisfies ϕ?

EX: $(\overset{T}{x_1}) \wedge (\overset{T}{x_1} \vee \overset{F}{\bar{x}_2}) \wedge (\overset{F}{\bar{x}_1} \vee \overset{T}{x_2} \vee \overset{F}{x_3} \vee \overset{F}{x_4} \vee \overset{F}{\bar{x}_5})$ ← let $x_3 = F$, $x_4 = F$, $\bar{x}_5 = F$

$\downarrow$ (same) $\downarrow$

$(\underset{T}{x_1} \vee \underset{F}{\bar{x}_1}) \wedge (\underset{T}{x_1} \vee \underset{F}{\bar{x}_2}) \wedge (\underset{F}{\bar{x}_1} \vee \underset{T}{x_2} \vee \underset{F}{x_3} \vee \underset{F}{x_4} \vee \underset{F}{\bar{x}_5} \vee \underset{T}{x_5} \vee \underset{T}{\bar{x}_4} \vee \underset{T}{\bar{x}_3} \vee \underset{T}{x_1} \vee \underset{F}{\bar{x}_2})$

Oracle Alg:

- Construct ψ from ϕ
  - → for every clause in ψ include the complement of all variables in the same clause of ϕ as well as the original variables (so for ex: $(\bar{x}_1 \vee x_2 \vee x_3 \vee x_4 \vee \bar{x}_5)$ in $\phi \rightarrow (\bar{x}_1 \vee x_1 \vee x_2 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_3 \vee x_4 \vee \bar{x}_4 \vee \bar{x}_5 \vee x$
- If ψ has a truth assignment θ s.t. ψ is satisfiable <u>and</u> θ satisfies the constraint that True must be assigned to exactly half the terms in each clause of ψ.

  Then output <u>yes</u>

  Otherwise, output <u>NO</u>