Homework #3:

1. max ?

s.t. (1) $x_1 \geq -4$

(2) $x_1 \leq 4$

(3) $x_2 \geq -4$

(4) $x_2 \leq 4$

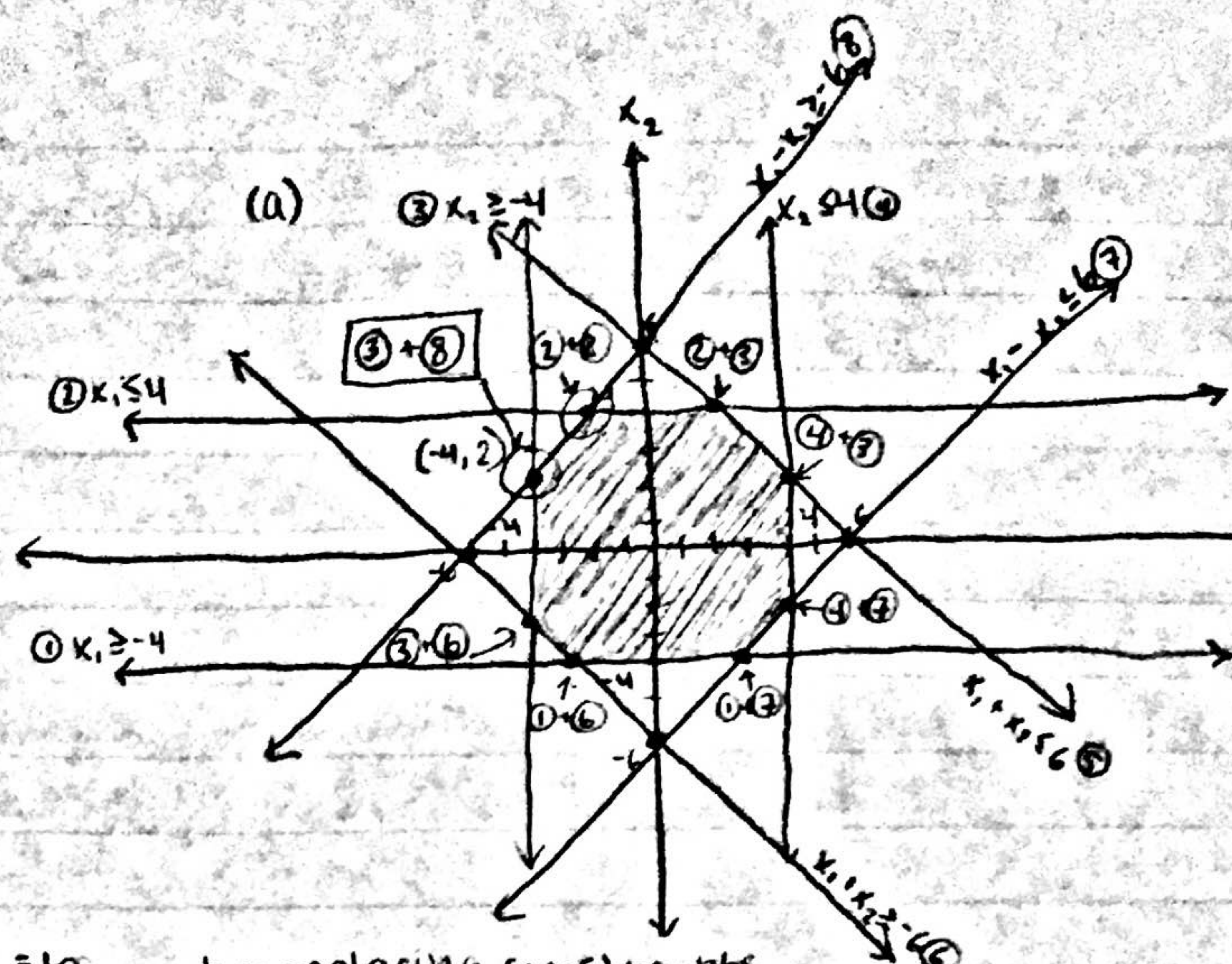
(5) $x_1 + x_2 \leq 6$

(6) $x_1 + x_2 \geq -6$

(7) $x_1 - x_2 \leq 6$

(8) $x_1 - x_2 \geq -6$

(a)



(3)+(8)

(b) $(-4,2) \rightarrow -2(-4)+2 = 10$ ... try replacing constraints to get adjacent points

(2)+(8) $= (-2,4)$ $-2(-2)+4 = 8$

(2)+(3) $= (2,4)$ $-2(2)+4 = 0$

(4)+(8) $= (4,2)$ $-2(4)+2 = -6$

(4)+(7) $= (4,-2)$ $-2(4)-2 = -10$

(1)+(7) $= (2,-4)$ $-2(2)-4 = -8$

(1)+(6) $= (-2,-4)$ $-2(-2)-4 = 0$

(3)+(6) $= (-4,-2)$ $-2(-4)-2 = 6$

So objective function = $\boxed{\max\ -2x_1 + x_2}$

(c) line segment between $(-4,2)$ and $(-2,4)$

...So between (3)+(8) and (2)+(8)

$(-1)(-4)+2=6$, $(-1)(-2)+4=6$

$(2,4) \rightarrow (-1)(2)+4 = 2$ $\quad (2,-4) \rightarrow (-1)(2)-4 = -6$

$(4,2) \rightarrow (-1)(4)+2 = -2$ $\quad (-2,-4) \rightarrow (-1)(-2)-4 = -2$

$(4,-2) \rightarrow (-1)(4)-2 = -6$ $\quad (-4,-2) \rightarrow (-1)(-4)-2 = 2$

So objective function = $\boxed{\max\ -1x_1 + x_2}$

2. LP with 5 variables and m linear constraints in canonical form

→ polynomial time algo $\quad O(5^m)$

- let 5 variables be x_1, x_2, x_3, x_4, x_5
- m linear constraints
- and an objective function (let be max of $c \cdot x$)

→ so, we iterate through each constraint one-by-one in order to find the optimal solution

→ when evaluating mth constraint:

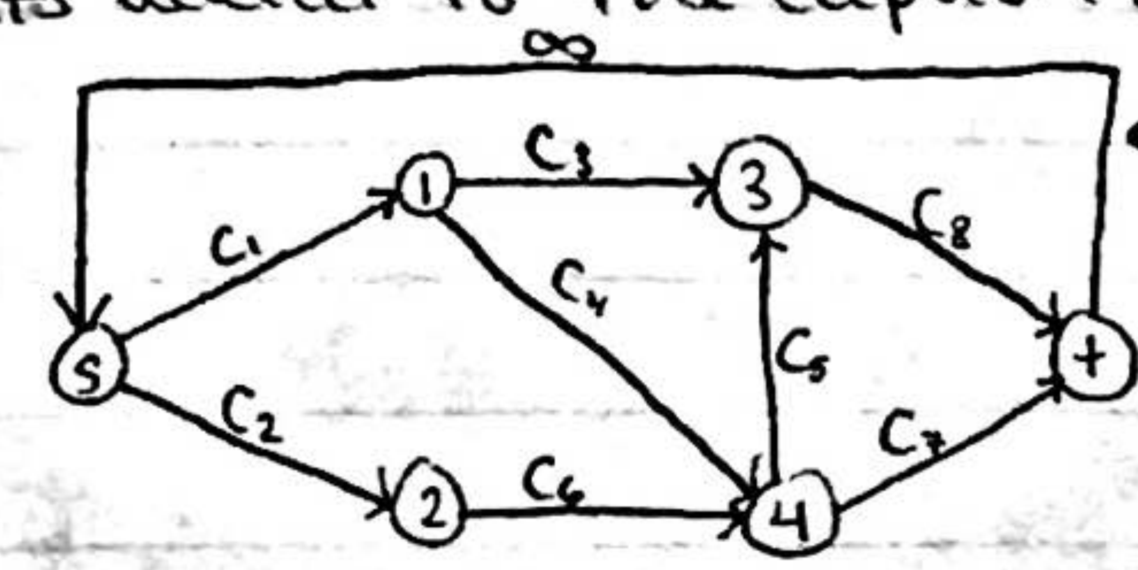
if current optimal solution satisfies the constraint, solution still optimal

else if current optimal solution does NOT satisfy contraint then the optimal solution will be transformed to be on the constraint's face of the feasible region in other words, this can be solved by graphing the feasible region, adding in the contraints one-by-one and finding the optimal solution to the constraints <u>so far</u>

But how can we show this is a polytime algorithm?
the run time of this algo is $O(m)$ because there are m constraints to evaluate, each taking $O(1)$ time

3. flow network $G=(V,E)$ s.t. $\{c_e\}_{e\in E}$, budget K for increasing capacities (we are allowed to increase the capacities of edges as long as the sum of the amounts added to the capacities is at most K)

let G be



← let this be c_9 (Let the flow be f_{ts})

$c_9 = c_1 + c_2$

$c_8 + c_7 = c_9$ (therefore we are trying to maximize the flow on e_9)

let the amount we are increasing each capacity be denoted by x_{uv}

max f_{ts}

s.t. $f^{in}(v) - f^{out}(v) = 0 \quad \forall v \neq s, v \neq t$

$f_{uv} \leq c_{uv} \quad \forall uv \in E$

$f_{uv} + x_{uv} \leq c_{uv} \quad \forall uv \in E$

$\sum_{uv\in E} x_{uv} \leq K \quad \forall uv \in E$

$f_{uv} \geq 0$

$x_{uv} \geq 0$

$c_{uv} \geq 0$

$\Rightarrow \sum_{u:(u,v)\in E} f_{uv} - \sum_{w:(v,w)\in E} f_{vw} = 0 \; \forall v \in V (v \neq s, v \neq t)$

So...

vars: $f_{uv} \quad \forall uv \in E$

$c_{uv} \quad \forall uv \in E$

$x_{uv} \quad \forall uv \in E$

K

max: f_{ts}

s.t.: $\sum_{u:(u,v)\in E} f_{uv} - \sum_{w:(v,w)\in E} f_{vw} = 0 \; \forall v \in V \, (v \neq s, v \neq t)$

$f_{uv} \leq c_{uv} \quad \forall uv \in E$

$f_{uv} + x_{uv} \leq c_{uv} \quad \forall uv \in E$

$\sum_{uv\in E} x_{uv} \leq K \quad \forall uv \in E$

$f_{uv} \geq 0$

$x_{uv} \geq 0$

$c_{uv} \geq 0$

$K \geq 0$

4. max $\frac{x_1+2x_2+3x_3+\dots+nx_n}{x_1+\dots+x_n}$ ← explain how can be solved using linear programming

s.t. $x_1+\dots+x_n \geq 0$

$\sum_{j=1}^{n} a_{ij}x_j \geq 0 \quad \forall i=1,\dots,m$ *objective function is division → must make it linear!

$x_i \geq 0 \quad \forall i=1,\dots,n$

max $\frac{x_1+2x_2+3x_3+\dots+nx_n}{x_1+\dots+x_n}$

s.t. $x_1+\dots+x_n < 0$

$a_{11}x_1+a_{12}x_2+a_{13}x_3+\dots+a_{1n}x_n \geq 0$

$a_{21}x_1+a_{22}x_2+a_{23}x_3+\dots+a_{2n}x_n \geq 0$

$\vdots$

$a_{m1}x_1+a_{m2}x_2+a_{m3}x_3+\dots+a_{mn}x_n \geq 0$

$x_1, x_2, x_3, \dots, x_n \geq 0$

So if we let $z = x_1+x_2+x_3+\dots+x_n = \sum_{i=1}^{n} x_i$ and let $y_i = \frac{x_i}{z}$ then we can represent the problem as follows...

max $y_1+2y_2+3y_3+\dots+ny_n$

s.t. $\sum_{j=1}^{n} a_{ij}y_j \geq 0 \quad \forall i=1,\dots,m$

$\sum_{i=1}^{n} y_i = 1$

$y_i \geq 0 \quad 1 \leq i \leq n$

5. $\max \sum_{P \in P} x_P$ ← sum of all paths

s.t. $\sum_{P: e \in P} x_P \leq c_e \quad \forall e \in E$ ← the sum of all paths P containing e are atmost c_e

$x_P \geq 0 \quad \forall P \in P$ ← all paths are non-negative

where $\sum_{P: e\in P}$ means that the sum is over all paths P that contain the edge e

(a) Why the following linear program solves the MAX-flow problem?

First of all, the objective function is trying to maximize the sum of all the paths. This is exactly what the max-flow problem is seeking to find as the flow of the graph is equal to the sum of the flow going through all the paths (and the max-flow is obviously seeking to maximize this). The second constraint is just noting that the flow of each path is non-negative. Finally, the first constraint is saying that the sum of all paths containing edge e are atmost c_e (the capacity of edge e). This is upholding the rules of a flow network given the capacity of each edge because the flow going through edge e <u>must</u> be less than or equal to the capacity of this edge. Furthermore, the total flow passing through the edge e must be equal to the sum of the flow of all the paths that pass through this edge. Therefore, the sum of the flow of all the paths P that contain the edge e ($\sum_{P: e \in P} x_P$) <u>must</u> be less than or equal to the capacity of e (c_e) for all edges in the graph. This explains why this linear program solves the MAX-flow problem.

proves that the feasible solution for the max-flow problem is of the same cost for the feasible solution of the LP

$\sum_{(u,v)\in E} f(u,v) = \sum_{P: e \in P} x_P = \sum_{w:(v,w)\in E} f(v,w)$

(b) Write the dual of the above linear program

$\max \sum_{P \in P} x_P$

s.t. $\left(\sum_{P: e \in P} x_P \leq c_e\right)^{\times y_e} \forall e \in E$

$x_P \geq 0 \quad \forall P \in P$

$\rightsquigarrow$

$$\boxed{\begin{array}{l} \text{vars: } \forall e \in E \ y_e \\ \text{min: } \sum_{e \in E} c_e y_e \\ \text{s.t. } \sum_{e \in P} y_e \geq 1 \ \forall P \in P \\ \qquad y_e \geq 0 \ \forall e \in E \end{array}}$$

(c) again... vars: $\forall e \in E \; y_e$

min: $\sum_{e \in E} c_e y_e$

s.t.: $\sum_{e \in P} y_e \geq 1 \quad \forall P \in \mathcal{P}$

$y_e \geq 0 \quad \forall e \in E$

- prove every s-t cut provides a feasible solution to the dual linear program s.t. value of dual linear program equals the capacity of the cut (thus, the solution to the dual LP is $\leq$ cap of min-cut)

this is saying that assigning a weight to each edge s.t. each s-t path is of a "distance" of at least 1... which can help us with min-cut

$\sum_{e \in E} c_e y_e$... if we let $y_e = 1$ if e is in the s-t cut (so let $e = uv$ s.t. $u \in A$ and $v \in B$) and if e is not in the s-t cut we let $y_e = 0$

therefore: $\sum_{e \in E} c_e y_e = \sum_{e \in s\text{-}t\,cut} c_e$ which equals the capacity of said s-t cut!

6. complementary slackness to show $x_1^* = x_3^* = 0.5$, $x_2^* = x_4^* = 0$, $x_5^* = 2$ is optimal!

$\max\ 3.1x_1 + 10x_2 + 8x_3 - 45.2x_4 + 18x_5$

s.t. $(x_1 + x_2 + x_3 - x_4 + 2x_5 \le 5) * y_1$ ①

$(2x_1 - 4x_2 + 1.2x_3 + 2x_4 + 7x_5 \le 16) * y_2$ ② *slack

$(x_1 + x_2 - 3x_3 - x_4 - 10x_5 \le -20) * y_3$ ③ *slack

$(3x_1 + x_2 + 3x_3 + \frac{3}{2}x_4 + \frac{7}{3}x_5 \le 10) * y_4$ ④ *slack. Where is the slack? → *2, *3, *4 constraints

$(x_2 + x_3 + 6x_4 + 2x_5 \le 4.5) * y_5$ ⑤

$+ (2x_2 - x_4 + x_5 \le 2) * y_6$ ⑥

$x_1, x_2, x_3, x_4, x_5 \ge 0$ → $y_2^* = 0$, $y_3^* = 0$, $y_4^* = 0$

$\min\ 5y_1 + 16y_2 - 20y_3 + 10y_4 + 4.5y_5 + 2y_6$

$y_1 + 2y_2 + y_3 + 3y_4 \ge 3.1$

$y_1 - 4y_2 + y_3 + y_4 + y_5 + 2y_6 \ge 10$

$y_1 + 1.2y_2 - 3y_3 + 3y_4 + y_5 \ge 8$

$-y_1 + 2y_2 - y_3 + \frac{3}{2}y_4 + 6y_5 - y_6 \ge -45.2$

$2y_1 + 7y_2 - 10y_3 + \frac{7}{3}y_4 + 2y_5 + y_6 \ge 18$

$y_1, y_2, y_3, y_4, y_5, y_6 \ge 0$

* if a dual var is > zero then the corresponding primal constraint must be an equality (tight)

*note: $x_1^*, x_3^*, x_5^* \ne 0$ → NO slack in 1st, 3rd, + 5th constraints in the dual

$y_1^* = 3.1$

$y_1^* + y_5^* + 2y_6^* \ge 10$

$y_1^* + y_5^* = 8$ → $y_5^* = 4.9$

$-y_1^* + 6y_5^* - y_6^* \ge -45.2$

$2y_1^* + 2y_5^* + y_6^* = 18$ → $(6.2) + (9.8) + y_6^* = 18$, $y_6^* = 2$

$y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^* \ge 0$

So... $y_1^* = 3.1$, $y_2^* = 0$, $y_3^* = 0$, $y_4^* = 0$, $y_5^* = 4.9$, $y_6^* = 2$

this is a feasible solution to the dual

$5(3.1) + 16(0) - 20(0) + 10(0) + 4.5(4.9) + 2(2)$

$15.5 + 22.05 + 4 = 41.55$ ← this is the cost of the dual

$3.1(0.5) + 10(0) + 8(0.5) - 45.2(0) + 18(2)$

$1.55 + 4 + 36 = 41.55$ ← this is the cost of the primal *cost dual = cost primal

⇒ weak duality says that 41.55 is optimal!

7. there are m raw materials and n mixtures, each composed of different proportions of these raw materials

for each $1 \leq i \leq n$, m numbers $a_{i1}, a_{i2}, \dots, a_{im} \in [0,1]$

→ a_{ij} fraction of the i-th mixture is from the raw material j

(m numbers add up to 1)

$u_i = \max_{j=1}^{m} a_{ij} - \min_{j=1}^{m} a_{ij}$ is the unbalancedness of the i-th mixture

prompt asks to explain how to solve using linear program (rather than formulate as linear program)

(x_i is how much of one mixture we want in our new one) ← so if x_i is 0.5 ⇒ 50% of new mix is of originally mix

(u_i corresponds to how unbalanced a mixture is (where 0 is perfectly balanced)

→ u_i is just a linear transformation for x_i ⇒ $\min u_i * x_i$

→ minimize the amount of unbalancedness (we want to have bigger fractions for the balanced u_i (closer to 0)