

# CS170Hw8

Trustin Nguyen

October 25, 2024

## Study Group

---

### Knightmare

---

Give a dynamic programming algorithm to find the number of ways you can place knights on an  $X$  by  $Y$  ( $X > Y$ ) chessboard such that no two knights can attack each other (there can be any number of knights on the board, including zero knights). Knights can move in a  $2 \times 1$  shape pattern in any direction.

**Provide a 4-part solution. Your algorithm's runtime should be  $O(2^{3Y}XY)$ , and return your answer mod 1789.**

### Max Independent Set Again

---

You are given a connected tree  $T$  with  $n$  nodes and a designated root  $r$ , where every vertex  $v$  has a weight  $W[v]$ . A set of nodes  $S$  is a  $k$ -independent set of  $T$  if  $|S| = k$  and no two nodes in  $S$  have an edge between them in  $T$ . The weight of such a set is given by adding up the weights of all the nodes in  $S$ , i.e.

$$W(S) = \sum_{v \in S} W[v]$$

Given an integer  $k \leq n$ , your task is to find the maximum possible weight of any  $k$ -independent set of  $T$ . We will first tackle the problem in the special case that  $T$  is a binary tree, and then generalize our solution to a general tree  $T$ .

- (a) Assume that  $T$  is a binary tree, i.e. every node has at most 2 children. Describe an  $O(nk^2)$  algorithm that solves this special case, and analyze its runtime. Proof of correctness and space complexity analysis are not required.

*Answer.* Run DFS on the graph, split the nodes into two groups, bipartite. Do an exhaustive search in each group

- (b) Now, consider any arbitrary tree  $T$ , with no restrictions on the number of children per node. Describe how we can add up to  $O(n)$  "dummy" nodes (i.e. nodes with weight 0) to  $T$ , as well as some edges, so that the resulting tree is a binary tree  $T_b$ .
- (c) Describe an  $O(nk^2)$  algorithm to solve the general case (i.e. when  $T$  is any arbitrary tree), and analyze its runtime. Proof of correctness and space complexity analysis are not required.

## Canonical Form LP

Recall that any linear program can be reduced to a more constrained *canonical form* where all variables are non-negative, the constraints are given by  $\leq$  inequalities, and the objective is the maximization of a cost function.

More formally, our variables are  $x_i$ . Our objective is  $\max(c^T x) = \max(\sum_i c_i x_i)$  for some constants  $c_i$ . The  $j$ th constraint is  $\sum_i a_{ij} x_i \leq b_j$  for some constants  $a_{ij}, b_j$ . Finally, we also have the constraints  $x_i \geq 0$ .

An example canonical form LP:

$$\begin{aligned} & \text{maximize } 5x_1 + 3x_2 \\ & \text{subject to } \begin{cases} x_1 + x_2 + x_3 & \leq 1 \\ -(x_1 + x_2 - x_3) & \leq -1 \\ -x_1 + 2x_2 + x_4 & \leq 0 \\ -(-x_1 + 2x_2 + x_4) & \leq 5 \\ x_1, x_2, x_3, x_4 & \geq 0 \end{cases} \end{aligned}$$

For each of the subparts below, describe how we should modify it to so that it satisfies canonical form. If it is impossible to do so, justify your reasoning.

Note that the subparts are independent of one another. Also, you may assume that variables are non-negative unless otherwise specified.

(a) Min Objective:  $\min \sum_i c_i x_i$

*Answer.* Change the min to a max and negate the cost function:

$$\max - \sum_i c_i x_i$$

(b) Lower Bound on Variable:  $x_1 \geq b - 1$

*Answer.* We have

$$x_1 - b + 1 \geq 0$$

So change of variables  $x'_1 = x_1 - b + 1$ . Now the condition is  $x'_1, x_2, x_3, x_4 \geq 0$ , and substitute all instances of  $x_1$  with  $x'_1 + b - 1$ .

(c) Bounded Variable:  $b_1 \leq x_1 \leq b_2$

*Answer.* Another change of variables:  $0 \leq x_1 - b_1 \leq b_2 - b_1$ . Have  $x'_1 = x_1 - b_1$ . Now the condition is  $x'_1, x_2, x_3, x_4 \geq 0$ , and substitute all instances of  $x_1$  with  $x'_1 + b_1$ . We also need to account for  $x'_1 \leq b_2 - b_1$ , so add that in as a new equation.

(d) Equality Constraint:  $x_2 = b_2$

*Answer.* Remove the constraint  $x_2 \geq 0$ , substitute all values of  $x_2$  in the LP to  $b_2$ . In your maximize function, you can just remove the term associated with  $x_2$  now because it is constant.

(e) More Equality Constraint:  $x_1 + x_2 + x_3 = b_3$

*Answer.* Add in the two constraints  $x_1 + x_2 + x_3 \leq b_3$  and  $x_1 + x_2 + x_3 \geq b_3$ .

(f) Absolute Value Constraint:  $|x_1 + x_2| \leq b_2$  where  $x_1, x_2 \in \mathbb{R}$

*Answer.* This means  $-b_2 \leq x_1 + x_2 \leq b_2$  or  $0 \leq x_1 + x_2 + b_2 \leq 2b_2$ . Then we can add in two constraints:

$$\begin{aligned}x_1 + x_2 + b_2 &\geq 0 \\x_1 + x_2 + b_2 &\leq 2b_2\end{aligned}$$

(g) Another Absolute Value Constraint:  $|x_1 + x_2| \geq b_2$  where  $x_1, x_2 \in \mathbb{R}$

*Answer.* It is required that  $x_1 + x_2 \geq b_2$  or  $x_1 + x_2 \leq -b_2$ . If we are allowed to create multiple LP problems, we can append  $x_1 + x_2 \geq b_2$  constraint to one LP problem and  $x_1 + x_2 \leq -b_2$  constraint to the same, but duplicated LP problem. Solve both, and return the one that gives the optimal solution between them. Otherwise, it is impossible to accommodate this because these two conditions are contradictory.

(h) Min Max Objective:  $\min \max x_1, x_2, x_3, x_4$

*Answer.* Consider the maximum on only two variables,  $x_1, x_2$ . This is given by

$$\frac{x_1 + x_2}{2} + \frac{|x_1 - x_2|}{2}$$

We can use another variable  $c_1 \geq \frac{|x_1 - x_2|}{2}$ . Recall that from part f, there is a canonical form fitting the absolute value constraint. Then the minimizing function becomes  $\min c_1 + \frac{x_1 + x_2}{2}$ . We can repeat this pairwise with the remaining  $x_3, x_4$ . At the end, turn the objective to maximizing the function by taking the negative of the loss function, and get rid of fractions by multiplying by some constant.

## Baker

You are a baker who sells batches of brownies and cookies (unfortunately no brookies... for now). Each brownie batch takes 4 kilograms of chocolate and 2 eggs to make; each cookie batch takes 1 kilogram of chocolate and 3 eggs to make. You have 80 kilograms of chocolate and 90 eggs. You make a profit of 60 dollars per brownie batch you sell and 30 dollars per cookie batch you sell, and want to figure out how many batches of brownies and cookies to produce to maximize your profits.

- (a) Formulate this problem as a linear programming problem; in other words, write a linear program (in canonical form) whose solution gives you the answer to this problem. Draw the feasible region, and find the solution using Simplex.

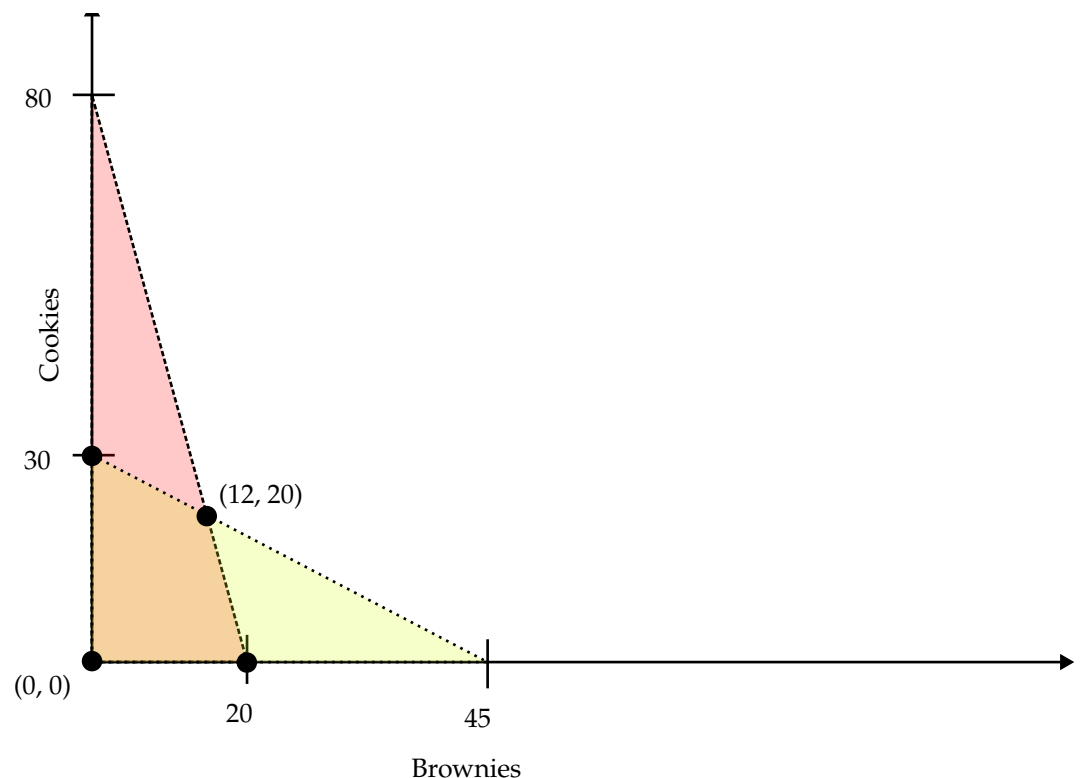
*Answer.* We have the table

	Chocolate	Eggs
Brownie	4	2
Cookie	1	3

Then if  $X$  is the number of brownies and  $Y$  is the number of cookies, then

$$\begin{cases} 4X + Y \leq 80 \\ 2X + 3Y \leq 90 \\ X \geq 0 \\ Y \geq 0 \end{cases}$$

Here is the region:



So we start at  $(20,0)$  and see which vertex gives the max profit, which is given by  $60X + 30Y = 30(2X + Y)$ . The current profit is  $30(40) = 1200$ . The vertex  $(12,20)$  gives:  $30(24 + 20) = 30(44) = 1320$ . The last vertex gives  $30(30) = 900$ . Then the max profit is from the batch of 12 brownies and 20 cookies.

- (b) Suppose instead that the profit per brownie batch is  $P$  dollars and the profit per cookie batch remains at 30 dollars. For each vertex you listed in the previous part, give the range of  $P$  values for which that vertex is the optimal solution.

*Answer.* The profit equation becomes

$$PX + 30Y$$

Plugging all of our batch numbers in:

$$(0, 0) \rightarrow (0 + 0)$$

$$(20, 0) \rightarrow (20P + 0)$$

$$(12, 20) \rightarrow (12P + 600)$$

$$(0, 30) \rightarrow (0P + 900)$$

For  $(0, 0)$  it is never optimal because the point  $(0, 30)$  will yield more profit for any  $P$ .

For  $(20, 0)$ , we require it to be more optimal than  $(12, 20), (0, 30)$ :

$$20P > 12P + 600 \implies 8P > 600 \implies P > 600/8 = 75$$

and

$$20P > 900 \implies P > 900/20 = 45$$

So  $P \geq 75$  for  $(20, 0)$ .

For  $(12, 20)$ :

$$12P + 600 > 20P$$

$$75 > P$$

$$12P + 600 > 900$$

$$12P > 300$$

$$P > 25$$

or  $20 \leq P \leq 75$ .

Finally for  $(0, 30)$ :

$$900 > 20P$$

$$45 > P$$

and from the last inequality,  $P < 25$ . So  $(0, 30)$  is optimal for  $P \leq 25$

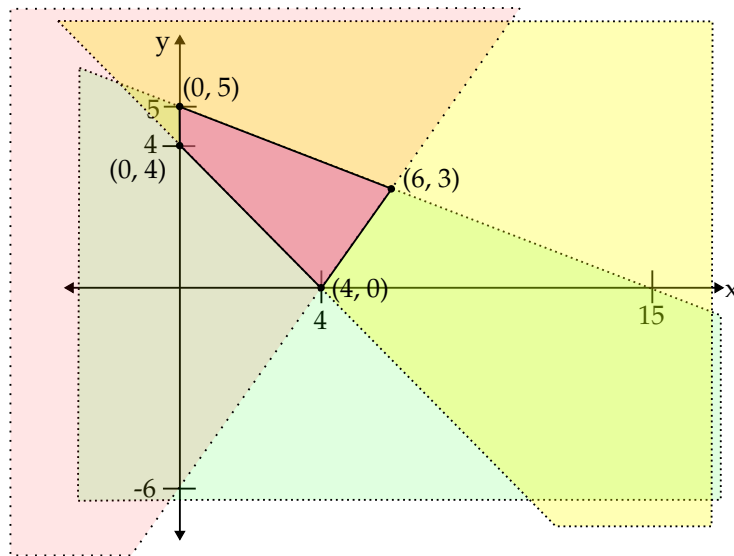
## Simply Simplex

Consider the following linear program:

$$\begin{aligned} & \max(5x + 4y) \\ & \text{subject to } \begin{cases} x + 3y \leq 15 \\ 3x - 2y \leq 12 \\ 4x + 4y \geq 16 \\ x \geq 0 \\ y \geq 0 \end{cases} \end{aligned}$$

(a) Sketch the feasible region. Make sure to clearly label your axes and vertices

*Answer.*



(b) Run the Simplex algorithm on this LP starting at  $(0, 4)$ . What are the vertices visited? **Please show your work.**

*Answer.* Starting at  $(0, 4)$ , the neighboring vertices are  $(4, 0)$  and  $(0, 5)$ . The objective function is  $5x + 4y$ . We evaluate at the three points:

$$(0, 4) \rightarrow 5(0) + 4(4) = 16$$

$$(0, 5) \rightarrow 5(0) + 4(5) = 20$$

$$(4, 0) \rightarrow 5(4) + 4(0) = 20$$

Since the two points are equal, consider  $(4, 0)$  first. The neighbor of  $(4, 0)$  is  $(6, 3)$ :

$$(6, 3) \rightarrow 5(6) + 4(3) = 42$$

It increased. Now we are back to  $(0, 5)$ , which results in a decrease from 42. So the  $x$  and  $y$  returned are  $(6, 3)$ . The vertices visited are  $(4, 0)$ ,  $(5, 0)$ ,  $(0, 4)$ ,  $(6, 3)$ .