15-281 Artificial Intelligence HW4

INSTRUCTIONS

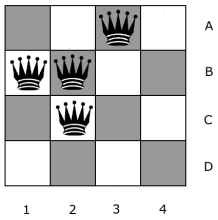
- **Due: Thursday, 13 February 2025 at 11:59 PM ET.** Remember that you may use up to 2 slip days for the Written Homework making the last day to submit **Saturday, 15 February 2025 at 11:59 PM ET**.
- **Format:** Write your answers in the `yoursolution.tex` file and compile a pdf (preferred) or you can type directly on the blank pdf. Make sure that your answers are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points. Handwritten solutions are not acceptable and may lead to lost points.
- **How to submit:** Submit a pdf with your answers on Gradescope. Log in and click on our class 15-281, click on the HW4 assignment, and upload your pdf containing your answers.
- **Policy:** See the course website for homework policies and academic integrity.

Name	
Andrew ID	
Hours to complete?	◯ (0, 2] hours ◯ (2, 4] hours ◯ (4, 6] hours ◯ (6, 8] hours ◯ > 8 hours

Q1. [11 pts] 4-Queens Problem

The following images show pseudocode for a hill-climbing algorithm to the 4-Queens problem and a starting 4-Queens problem state. The queens move normally as they do in chess (ie. they may move as many spaces as they want in each direction as long as their path is not blocked).

```
function 4-QUEENS-HILL-CLIMBING(problem)
    current_state = problem.INITIAL-STATE
    loop do
        neighbor = state with lowest number of conflicts between queens obtained by
                   making one legal chess move from current_state
        if neighbor.num_conflicts >= current_state.num_conflicts
            return current_state
        current_state = neighbor
```



(a) [5 pts] Write down the states of each of the 4 queens after one iteration of the loop. Note that one iteration of the loop is equivalent to moving one queen.

States:

(b) [6 pts] Will the hill climbing algorithm presented above stop for the given starting state? If so, what does the state end up as? Please check all that apply.

- ☐ A) Will not stop
- ☐ B) Local optima
- ☐ C) Global optima
- ☐ D) Shoulder
- ☐ E) None of the above

Q2. [22 pts] Pinky's Day

Pinky is trying to organize her time in a day. Except for sleeping, eating, and (most importantly) posting on Instagram (@pinkythepenguin15281), she still has 8 hours to spare each day. Two important things in her life are *partying* and *doing homework*. In order to keep up with her classes, she has to spend ***at least*** 2 hours per day on homework. But she doesn't want the wear herself out, so the number of hours she spends partying should equal ***at least*** half the number of hours she spends on homework. On the other hand, she feels bad if she parties too much, so her party time ***should not exceed*** her homework time by more than 3 hours.

Pinky knows that she will gain 1 unit of happiness for every hour she parties, and she will loose 0.5 unit of happiness for every hour she spends on homework. But she needs your help to figure how much time she should spend on partying and how much time she should spend doing homework such that her happiness is **maximized**.

(a) [9 pts] Write this problem as an LP in **inequality form** as defined in lecture. Define variable x_1 to be hours spend in parties, and variable x_2 to be the time spend on homework. *Warning:* Be sure to strictly follow the inequality form, including the proper use of less than or equal, or you will lose points.

Inequality Form:

(b) [9 pts] Accurately plot the graphical representation of this linear program. Specifically:

- Plot the boundary of each halfspace as a line (no need to shade or draw normal vectors), and
- Plot the cost vector as an arrow with *magnitude one*, somewhere within the feasible region.

Do *not* draw; use a plotting tool such as Python matplotlib and include the resulting image here. Be sure to label the axes of your plot, including tick-marks. Display your plot with a *square* aspect ratio, e.g. in matplotlib: `plt.axis("equal")`. Additionally, zoom your plot to make the entire feasible region visible.

Tip to a plot vector $[v_1, v_2]^T$ in matplotlib starting at some point (x_1, x_2):

```
plt.quiver(x1, x2, v1, v2, angles="xy", scale_units="xy", scale=1)
```

Tip to properly control scaling using a specific width and height:

```
plt.figure(figsize=(width,height))
```

We have included some starter code for you in `figures/plot_graph.py`. You will need to modify the `plot_graph()` function and fill in code for `compute_unit_length()` yourself.

Note: For the sake of grading please let your x be of range $[-2, 10]$ and y be of range $[-4, 8]$.

Plot:

(c) [4 pts] Find the optimal solution to the LP problem. Give the solution hours she should spend in party and homework respectively and her happiness.

Party:	**Homework:**	**Happiness:**

Q3. [16 pts] Graphing LPs

For the inequality form of a linear program, and a given A matrix and $\mathbf{b}$ vector,

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \ & A\mathbf{x} \preceq \mathbf{b} \end{aligned}$$

For each row i of A and $\mathbf{b}$, accurately plot 1) the line $a_{i,1}x_1 + a_{i,2}x_2 = b_i$ and 2) the vector $[a_{i,1}, a_{i,2}]^T$ as an arrow beginning at any point on its respective line with *magnitude one*.

Tip to plot a vector $[v_1, v_2]^T$ in matplotlib starting at some point (x_1, x_2):

```
plt.quiver(x1, x2, v1, v2, angles="xy", scale_units="xy", scale=1)
```

Tip to properly control scaling using a specific width and height: `plt.figure(figsize=(width,height))`

Do *not* draw; use a plotting tool such as Python matplotlib and include the resulting image here. Be sure to label the axes of your plot, including tick-marks. Display your plot with a *square* aspect ratio, e.g. in matplotlib: `plt.axis("equal")`. Additionally, zoom your plot or adjust the axes such that all of the vectors are visible. You do not need to shade the feasible regions.

We have included some starter code for you in `plot_graph.py`. You will need to modify the `plot_graph()` function and fill in code for `compute_unit_length()` yourself.

(a) [8 pts]

$$A = \begin{bmatrix} 3 & 5 \\ 7 & 6 \\ 12 & 6 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 10 \\ 17 \\ 27 \end{bmatrix}$$

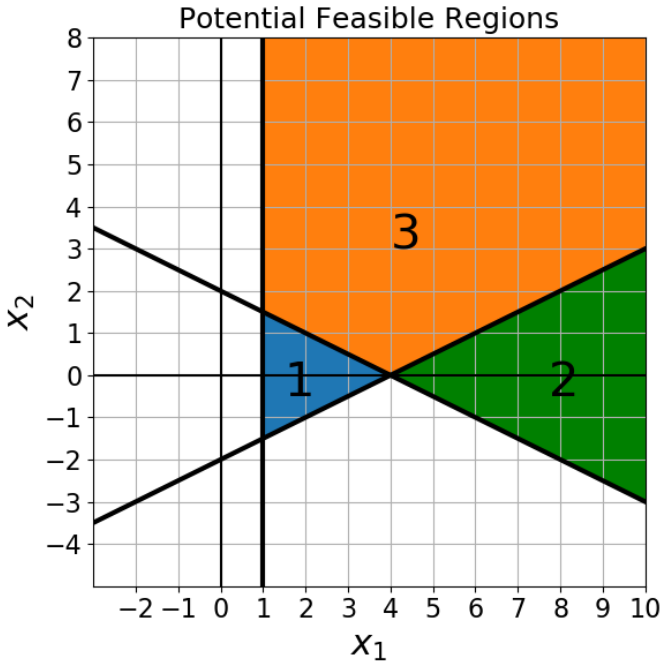
Plot:

(b) [8 pts]

$$A = \begin{bmatrix} -2 & -1 \\ 2 & 5 \\ 7 & 2 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -4 \\ 10 \\ 11 \end{bmatrix}$$

Plot:

Q4. [18 pts] Feasible Regions



In this problem, you are given a graph with constraint boundary lines (**bolded**) and potential feasible regions. You may assume shaded regions at the edge of the plot continue to infinity. Provide a corresponding A and $\mathbf{b}$ using the inequality form below which defines each feasible region in the boxes below. You must include all constraints.

$$\min_{\mathbf{x}} \ \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} \preceq \mathbf{b}$$

(a) [6 pts] Feasible Region 1

A:

$\mathbf{b}$:

(b) [6 pts] Feasible Region 2

A:

$\mathbf{b}$:

(c) [6 pts] Feasible Region 3

A:

$\mathbf{b}$:

Q5. [23 pts] Integer Programming

Stock portfolio investment is a common application of linear and integer programming. Suppose you are interested in investing in two companies, PinkyCorp and Honk Inc. Let x_1 represent the number of shares of PinkyCorp stock you purchase and x_2 be the number of shares of Honk Inc. A good way to hedge your bets is to diversify your portfolio, so you want at least 2 shares of PinkyCorp's stock and at least 2 shares of Honk Inc's. One share of PinkyCorp costs \$1, and one share of Honk Inc. costs \$2. You want to spend at most \$10. You also want to keep your risk below some risk threshold R. Each share of PinkyCorp has risk 3, and each share of Honk Inc. has risk 1. How much of each stock should you buy to maximize profits if PinkyCorp has profit \$4 per share, and Honk Inc. has profit \$1 per share.

(a) [3 pts] Write the A, $\mathbf{b}$, and $\mathbf{c}$ matrix in inequality form (minimization). Assume $\mathbf{x} = [x_1, x_2]^T$ in that order. Use parameter R if needed.

A:	**b:**	**c:**

(b) [4 pts] If $R = 13$, what are the corners (x_1, x_2) constrained polygon?

Corners:

(Optional) For partial credit on (b) and (c), you may choose to show your work below:

(c) [2 pts] What are the coordinates (x_1, x_2) of the optimal objective? What is the optimal objective value of this minimization problem?

LP Solution Coordinates:	**Profit:**

(d) [8 pts] You should find that the minimum cost corner is a non-integer. On your stock trading site, you must buy whole shares of stock. Run branch and bound to find the optimal number of integer shares of stock you should buy. Follow these instructions very carefully:

- Each iteration corresponds to that depth of the branch and bound tree. Fill in the appropriate information for each iteration until the branch and bound algorithm finishes and returns. *Use **only** as many rows the algorithm needs before it returns.*
- Specify **ALL** the constraints you add to the problem *in addition to the original set.* For example, if you first branch on $x_2 \leq 10$ and $x_2 \geq 11$, then you'd write for the left branch $x_2 \leq 10$ and for the write branch $x_2 \geq 11$. If you then pull $x_2 \geq 11$ off the queue and branch on $x_1 \leq 20$ and $x_1 \geq 21$, the second iteration would have left branch "$x_2 \geq 11, x_1 \leq 20$" and right branch "$x_2 \geq 11, x_1 \geq 21$"
- If a particular solution is infeasible, write "*infeasible*" in that box.

Iteration	Left Branch Constraints	Left Solution (x_1, x_2)	Right Branch Constraints	Right Solution (x_1, x_2)
1	(i)	(ii)	(iii)	(iv)
2	(v)	(vi)	(vii)	(viii)
3	(ix)	(x)	(xi)	(xii)
4	(xiii)	(xiv)	(xv)	(xvi)

What is the optimal objective and its value of the integer solution of this minimization problem?

IP Solution Coordinates:	**Profit:**

(e) [6 pts] You decide that you're willing to be riskier in your stock purchases in order to make more money. Do not change any other constraints except the value of risk R. Find the minimum value of R that maximizes the total profit given the rest of the constraints.

Minimum R:	**New IP Solution:**	**New Profit:**

Q6. [10 pts] Ethical Considerations for Amazon Delivery Routes

The following question will be about *Amazon's Cost Saving Routing Algorithm Makes Drivers Walk Into Traffic*: `https://www.vice.com/en/article/amazons-cost-saving-routing-algorithm-makes-drivers-walk-into-traffic/`

(a) [4 pts] Give two possible parameters that the Amazon delivery routing algorithm currently optimizes for. Why do you think these are being used?

Answer:

(b) [3 pts] Why are these parameters in conflict with the delivery drivers' needs? Additionally, give two possible parameters that the routing algorithm could add to make it more fair for the drivers.

Answer:

(c) [3 pts] Mike, the anonymous interviewee, gave a lot of information about his route and troubles in the interview. Do you think it would be ethical for Amazon, using their data on delivery routes, to design an algorithm to search for who Mike really was? Why or why not?

Answer: