1 Linear Programming (Conceptual Practice)

Consider a linear program with a singular constraint of the form $a_1x_1 + a_2x_2 \leq b$ and cost vector $\boldsymbol{c} = \begin{bmatrix}-2 & 3\end{bmatrix}$. You are given the following list of possible points (x_1, x_2) to consider:

$$(5,-5),(-5,5),(0,5),(5,0),(-5,-5),(5,5)$$

1. Which of the points would maximize the objective value? Which of the points would minimize the objective value?

2. Suppose you are given $\boldsymbol{a} = \begin{bmatrix}5 & -4\end{bmatrix}$. What are the values $a_1x_1 + a_2x_2$ for each of the points? If $b = 9$, which points are feasible?

3. With the same $\boldsymbol{a}$ from 2., which of our points is feasible and minimizes total cost? Going outside the given list of points, how would we decrease total cost? What is our **minimum** objective value?

4. Now with $\boldsymbol{a} = \begin{bmatrix}2 & -3\end{bmatrix}$, how would we decrease total cost? What is our **minimum** objective value?

5. In general, when would minimizing LPs with a singular constraint not have a minimum objective at $-\infty$?

2 CSPs (Arc Consistency, Search)

You've generously saved a row of 6 seats in Rashid for 6 of your 15-381 classmates (A-F[1]), and are now trying to figure out where each person will be seated. You know the following pairs of people have some kind of binary constraint between them:

- A, B
- A, C
- A, D
- B, C
- B, E
- C, F

1. Draw the constraint graph to represent this CSP.

2. Some value is assigned to A. Which domains could change as a result of running forward checking for A?

3. Some value is assigned to A, and then forward checking is run for A. Then some value is assigned to B. Which domains could change as a result of running forward checking for B?

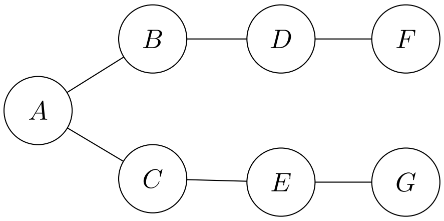
4. Some value is assigned to A. Which domains could change as a result of enforcing arc consistency after this assignment?

5. Some value is assigned to A, and then arc consistency is enforced. Then some value is assigned to B. Which domains could change as a result of enforcing arc consistency after B's assignment?

6. You are now trying a brand new algorithm to solving CSPs by enforcing arc consistency initially, then **after every even-numbered assignment** of variables (after assigning 2 variables, then after 4, etc.).

   You have to backtrack if, after assigning a value to variable X, there are no constraint-satisfying solutions. Mathematically, for a single variable with d values remaining, it is possible to backtrack d times in the worst case. For each of the following constraint graphs, assume each variable has domain of size d. How many times would you have to backtrack in the worst case for the specified orderings of assignments?

[1] not correlated with their grades



(a) ABCDEFG:

(b) GECABDF:

(c) GFEDCBA:

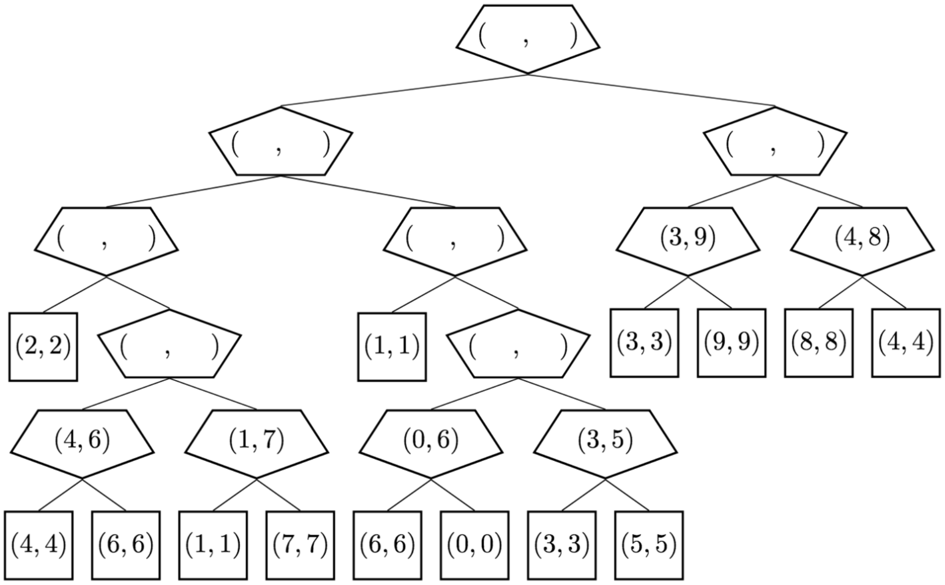
3 Games (Pruning)

Consider a two-player game between Pacman and a ghost (Casper) in which both agents alternate moves. As usual, Pacman tries to maximize his own utility. Unlike the usual minimizer ghost, Casper is friendly and **helps Pacman** by maximizing his utility. **Pacman is unaware** of this and acts as if Casper is playing against him. **Casper knows Pacman is misinformed** and acts accordingly.

1. For the game tree for this game, the value pair at each node is (u, v), where u is the value of the subtree as determined by Pacman, and v is the value of the subtree as determined by Casper.
   For terminal states, $u = v =$ the utility of that state.

   For example, in the subtree below with values (4, 6), Pacman believes Casper would choose the left action (with value = 4), but Casper actually chooses the right action (with value = 6), since that is better for Pacman.

   Fill in the remaining (u, v) values in the tree for this game below, where Casper is the root. Casper nodes are upside-down pentagons, and Pacman nodes are rightside-up pentagons.



2. In the game tree above, put an 'X' on the branches that can be pruned and do not need to be explored when **Casper** computes the value of the tree. Assume that children are visited left-to-right.

3. What would the value of the entire game tree be if Pacman knew that Casper is friendly?

4 Search (Algorithms & Properties)

1. When can we guarantee completeness and optimality (if ever) for each of the following search algorithms we've seen in class? For each algorithm, indicate under what conditions it is complete and/or optimal.

Algorithm	**Complete**	**Optimal**
Breadth-First		
Depth-First		
Iterative Deepening		
A*		

2. Consider a dynamic A* search where after running A* graph search and finding an optimal path from start to goal, the cost of one of the edges $X \to Y$ in the graph changes. Instead of re-running the entire search, you want to find a more efficient way of returning the optimal path for this new search problem.

   For each of the following changes, describe how the optimal path cost would change (if at all). If the optimal path itself changes, describe how to find the new optimal path. Denote c as the original cost of $X \to Y$, and assume $n > 0$.

   (a) c is increased by n, $X \to Y$ is on the optimal path, and was explored by the initial search.

   (b) c is decreased by n, $X \to Y$ is on the optimal path, and was explored by the initial search.

   (c) c is increased by n, $X \to Y$ is not on the optimal path, and was explored by the initial search.

   (d) c is decreased by n, $X \to Y$ is not on the optimal path, and was explored by the initial search.

   (e) c is increased by n, $X \to Y$ is not on the optimal path, and was not explored by the initial search.

   (f) c is decreased by n, $X \to Y$ is not on the optimal path, and was not explored by the initial search.

5 Agents and Environments

Consider the following 3 agent classes:

```
class A:
    def act(percept):
        return fA()
```

```
class B:
    def act(percept):
        return fB(percept)
```

```
class C:
    percepts, initially []
    def act (percept):
       push(percepts, percept)
       return fC(percepts)
```

In each of the agents, the function *f* is some arbitrary, possibly randomized, function of its inputs with no internal state of its own.

1. For each of the following sets of properties, list the agents for which there exists some *f* such that the agent is perfectly rational in every possible environment with those properties.

   (a) Fully observable, deterministic, discrete, single-agent, static

   (b) Partially observable, deterministic, discrete, single-agent, static

   (c) Partially observable, stochastic, discrete, single-agent, static

2. **True/False:** There exists an environment in which every agent is rational. Describe such an environment, or explain why it cannot exist.