MATH 371 - Game Theory

Justin Overstreet, Austin Spurlin, and Jonas Smith

Spring 2020

For the Game Theory Project we will be using python, and several different libraries to solve our problem.

We first load the "game" from a user defined excel sheet.

```
# Read information from user defined excel sheet
excel_file = ask_for_file()
game_info = pd.read_excel(excel_file)
```

Next we define the Game that we will use with our Dominance function to find dominance

```
# Game Definition Section
cols = np.array(game_info.iloc[:, 0].tolist())
game_info.drop(game_info.columns[0], axis=1, inplace=True)
rows = np.array(game_info.columns.tolist())
game = np.asarray(game_info)
initialGame = game
```

In our dominance function we first iterate through the rows and columns and determine if dominance can found. We do a very similar process as below for columns.

```
    # Loop through each row, looking for another row that is always worse.
    remRows = []
    for i in range(0, game.shape[0]):
        for j in range(0, game.shape[0]):
            if i == j or j in remRows:
                continue
            rem = True
            for k in range(0, game.shape[1]):
                if game[j][k] > game[i][k]:
                    rem = False
                    break
            if rem:
                remRows.append(j)
```

We then remove the non-dominant rows/cols...

```
    # Remove the rows.
    for i in remRows:
        game = np.delete(game, i, 0)
        rows = np.delete(rows, i, 0)
    # Remove the columns.
    for i in remCols:
        game = np.delete(game, i, 1)
        cols = np.delete(cols, i, 0)
```

We now call our methods that will build the linear equations for Player 1, and Player 2, and solve the system of linear equations.

```
    p1 = PrettyTable(cols)
            p1.title = "Player 1"
            choices = optimizePlayer1(game)

            for i in range(0, len(choices)):
                choices[i] = f"{choices[i]:.2%}"
```

Lastly we print our solution...

```
        print(p1)
        print(p2)
```