**Adversarial Search Project**

Implement the following four functions:

**1. minimax.** Implement the minimax algorithm for two-player, constant-sum games. Recall from the game theory lecture that constant-sum describes games in which the sum of the rewards earned by the players is the same across all possible outcomes. Your implementation will search all the way down to terminal states.

**2. alpha\_beta.** Same as part 1 but use alpha-beta pruning.

**3. alpha\_beta\_cutoff.** Same as part 2, but this time cutting off search after some given number of turns and applying a given evaluation function. See the function’s docstring for more details.

**4. general\_minimax.** A generalization of minimax (no pruning) that supports games with two or more players and arbitrary (potentially variablesum) reward structures and chooses a maximin action. Read the subsection below for details.

For the first three coding tasks (but not the fourth), you may assume that players’ turns are alternating. This means that the second player always takes his turn after the first player, and the first player always takes her turn after the second player. You may never assume that you are the first player, but 1 you should always maximize your score.

Each function that you will implement takes in an AdversarialSearchProblem as an input—much like how your search algorithms took in a SearchProblem in the previous assignment—and outputs an action to take. We strongly recommend reading and understanding adversarialsearchproblem.py before writing any code.

**Minimax for Variable-Sum Games**

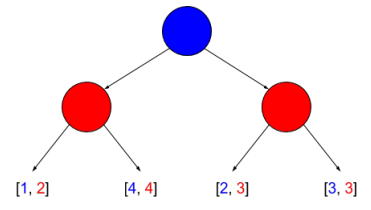
**The Goal:** The goal of each agent is still to maximize his own points, irrespective of the points that the other agents earn. That is, as Player 1, you should prefer [3, 100] to [2, 0].

**The Algorithm:** It is not clear how to generalize the minimax algorithm from the constant-sum setting to the variable-sum setting. There at least two reasonable approaches: the faithful approach and the conservative approach.

1. **Faithful Approach:** Prescribe the rational action assuming Common Knowledge of Rationality (CKR) between the players. Recall from the game theory lecture that CKR means that every player is rational, and every player knows that every player is rational, and every player knows that every player knows that every player is rational, and so on, ad infinitum. Here, “rational” describes a player who chooses the action to maximize their points given their beliefs about what the other players will do.

2. **Conservative Approach:** Prescribe the maximin action. Recall from the game theory lecture that the maximin action is the one that ensures the best worst-case scenario for the actor.

Here is an example that illustrates the difference between the faithful and conservative approaches in this setting.



In the diagram above, there are just two players: Blue and Red. The lists at the bottom represent terminal states of the game, in which the first number is Blue’s points, and the second number is Red’s points.

The Blue player can move Left or Right.

If Blue takes the faithful approach, Blue thinks about what Red would do in each subgame, assuming that Red is rational. If Blue were to move Left, then a rational Red would move Right, earning payoff [4, 4] for the players, and if Blue were to move Right, then a rational Red would assess Left and Right as equally favorable options, earning payoff [2, 3] or [3, 3] for the players. After completing this analysis, Blue would decide to move Left, since doing so would lead to more points under the assumption of CKR.

In contrast, if Blue takes the conservative approach, what would Blue do? If Blue moves Left, then the worst-case scenario earns 1 point, and if Blue moves Right, then the worst-case scenario earns 2 points. Trying to make the best worst-case scenario, the conservative approach prescribes Right in this case.

The generalization of minimax that you will implement (for general\_minimax in the stencil) will take the conservative approach and choose a maximin action.