Homework 3: Adversarial Search

In homework 3, you will implement an "AI" to play two-player Sweep Fives. The instructions for this game were sent out separately.

You should not discuss this homework with anyone other than the TA and instructor. Make sure to include this statement in your submission:

On my honor, as a UNT student, I have neither given nor received unauthorized assistance on this work.

(Later problem instructions will be elaborated over the course of the next week.)

1. **Legal Actions Generation (5 points):** A key part of implementing an adversarial search is to generate the legal actions (or plays). In a simple depth first search, these actions could potentially be generated one at a time and a given action could be immediately explored before the next action is generated. However, in class we saw that the total search space is enormous, so we will eventually want to guide the search in a way that optimizes our results for a given amount of search time. In order to accomplish that, we first want to generate all of the legal actions, and then we can use a heuristic to order their exploration advantageously.

   Update the actions() function in the code provided to you for this homework, such that it returns a list of lists, where each embedded list represents a legal action (a play the agent can make) and the set of all of those lists is complete (it includes every possible action that is legal according to the rules of the game; the list should not just be the single best play – that will be selected as part of the implementation of the Minimax algorithm in a subsequent part of this homework).

   The key data structures required for this task are the agent’s (player’s) hand and the table (the dominos face-up on the table). Each of these is represented as a list of tuples (specifically, pairs), where a pair represents a single domino (or card) by indicating the two pip counts of the domino, ranging from zero to six. See the code for details.

   Each action list should start with the pair representing the domino from the player’s hand that is to be played. When the action is a discard, this will be the only pair in the list. When the domino is being used to take other dominos from the table, the card from the hand should be followed in the list by n pairs representing the n dominos to be taken from the table.

   Test your revised GenerateActions.py file using the test harness and cases provided.

Submit your revised GenerateActions.py file to Nishitha and I before class this Thursday, February 16.

2. **Minimax Algorithm (20 points): This question will be fleshed out in the next couple of days.** Normally Sweep Fives is not a fully observable game, but in this question, we will treat it as such. Follow all of the normal rules of sweep fives except, assume there are only two players and that the game consists of a single hand (three dominos), which
are all dealt face up (you can observe your opponent’s hand). Implement the Minimax algorithm to play an optimal game strategy for a single hand.

3. *Alpha-beta Pruning (20 points):*

4. *Stochastic Scenario (20 points):*

5. *Partially Observable Scenario (20 points):*