Heuristic Results:

The results for the ID_Improved player during tournament.py:

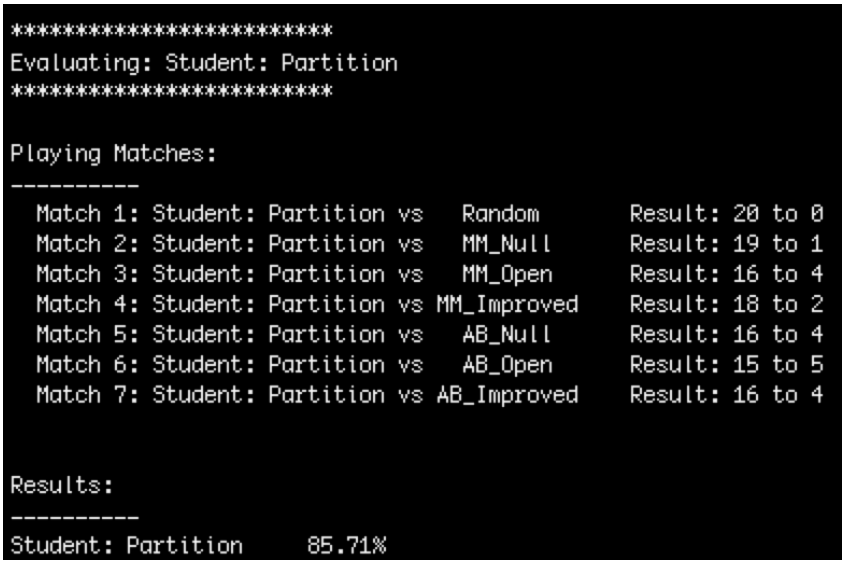
```
************************
 Evaluating: ID_Improved
************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 19 to 1
  Match 2: ID_Improved vs   MM_Null     Result: 18 to 2
  Match 3: ID_Improved vs   MM_Open     Result: 15 to 5
  Match 4: ID_Improved vs MM_Improved   Result: 15 to 5
  Match 5: ID_Improved vs   AB_Null     Result: 18 to 2
  Match 6: ID_Improved vs   AB_Open     Result: 14 to 6
  Match 7: ID_Improved vs AB_Improved   Result: 12 to 8


Results:
----------
ID_Improved         79.29%
```

I tested three custom heuristics using tournament.py:

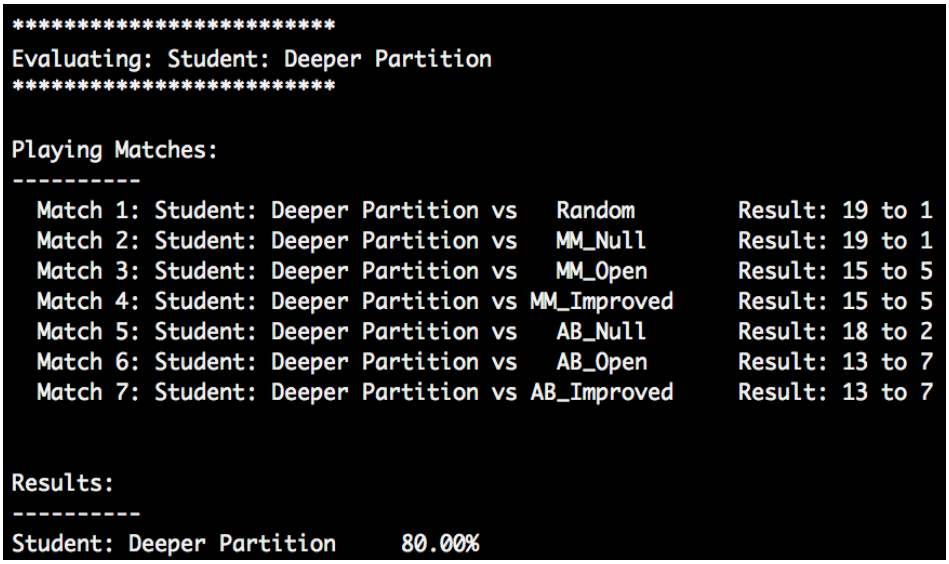
1. Deeper: Modified the Improved heuristic so that for each possible move in improved, Deeper forecasted the game board and calculated the number of possible moves on that board. In this way, positions towards the middle of the board are given a higher weight since they are more likely to have more moves available in two turns.

```
************************
Evaluating: Student: Deeper
************************

Playing Matches:
----------
  Match 1: Student: Deeper vs   Random          Result: 19 to 1
  Match 2: Student: Deeper vs   MM_Null         Result: 18 to 2
  Match 3: Student: Deeper vs   MM_Open         Result: 18 to 2
  Match 4: Student: Deeper vs MM_Improved       Result: 14 to 6
  Match 5: Student: Deeper vs   AB_Null         Result: 18 to 2
  Match 6: Student: Deeper vs   AB_Open         Result: 14 to 6
  Match 7: Student: Deeper vs AB_Improved       Result: 12 to 8


Results:
----------
Student: Deeper     80.71%
```

2. Partition: Determines whether a board has a vertical or horizontal partition of two or more squares, then calculates the number of blank spaces on each side of the partition. Since players cannot jump the partition, positions on the side with more blank spaces are more valuable.

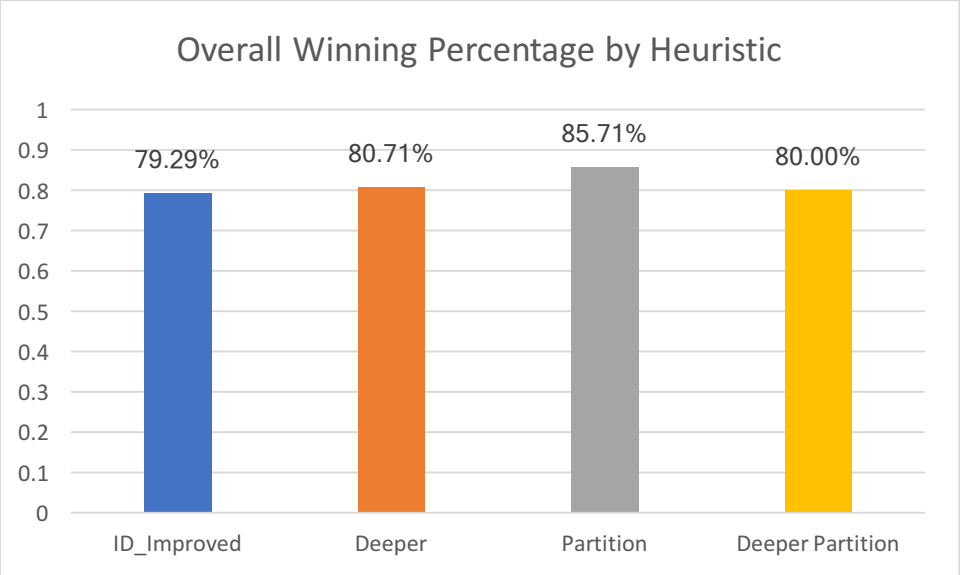


3. Deeper Partition: Combines the previous two heuristics, calculating scores based on the number of moves available two moves away, and then multiplying by a multiplier if the board has a partition.



Heuristic Analysis and Comparison:

The winning percentages for the various heuristics are displayed in the chart below:



The Deeper and Deeper Partition heuristics performed slightly better than the ID_Improved player for the given tournament size. It is unlikely that this difference is statistically significant. The Partition heuristic performed best. In particular, it performed better than all others against the AB_Open and AB_Improved players. It is possible that Deeper Partition did not improve on Partition because of the increased time to evaluate board states may have reduced the time available to search in depth.

It is likely that the effectiveness of Partition would improve given a larger board. On a 7x7 board, a partition is only likely to appear later in the game when there are few moves left, mitigating the advantage that a partition can impart. In addition, only strictly horizontal and vertical were considered. An improved heuristic would take into account non-linear partitions. There is also room to experiment with the multiplier to determine an optimal premium for moving to the more spacious side of a partition.

Heuristic Recommendation:

I recommend adopting the partition heuristic. The partition is not as advantageous in the "knight" variant of isolation since a one-square partition can be hopped, and a two-square partition is more difficult to create. Nevertheless, the partition heuristic exhibited a superior winning percentage to the ID_Improved player. In addition, it incorporates information about the board state beyond the availability of moves, which reflects the fact that partitions can confer significant advantages. Finally, by maintaining the simplicity of the Improved heuristic, Partition avoids spending too much time calculating a value, as may be the case with the Deeper Partition. For these three reasons, I recommend adopting the Partition heuristic.