Project 2 - Game playing : Adversarial search for the game "Isolation"

1. **Description:**
   - Adversarial games are defined as games where win of one player is a loss of the other
   - States: board configurations
   - Initial state: the board position and which player will move
   - Successor function: returns list of (move, state) pairs, each indicating a legal move and the resulting state
   - Terminal test : determines when the game is over
   - Utility function: gives a numeric value in terminal states (e.g., -1, 0, +1 for loss, tie, win). Adversarial games i.e win of one player is a loss of the other

2. **Heuristic_score_player_moves:**
   This is the baseline scoring system when the agent optimizes its moves.
   - Result :
     i. ID_Improved: 80.00%
     ii. Student : 77.14%
     iii. Different : -2.86%

3. **Heuristic_score_normal_difference_moves:**
   This is the baseline scoring system when the agent has more moves than opponent
   - Result :
     i. ID_Improved: 77.86%
     ii. Student : 77.86%
     iii. Different : 0%

4. **Heuristic_score_quadratic_difference_moves:**
   This heuristic take into account of the difference of squared number of legal moves of player and opponent.
   - Result :
     i. ID_Improved: 81.43%
     ii. Student : 76.43%
     iii. Different : -5%

5. **Heuristic_score_final_custom_score:**
   This heuristic assign various weights to 3 heuristics (heuristic_score_player_moves, heuristic_score_normal_difference_moves, heuristic_score_quadratic_difference_moves) to see the effect of different heuristics.
   - Result :
     i. ID_Improved: 86.43%
     ii. Student : 79.29%%
     iii. Different : -7.15%

6. ***Analysis:***

The heuristic_score_normal_difference_moves is chosen because it produces the max score . This heuristics values the moves that increase the availability of spaces and minimize the component's moves. Another strategy is setting optimal set of weights for each heuristic can improve the performance when the game will change.