**Isolation Game playing agent Heuristics Performance Analysis**

While playing game of isolation on a board bigger than 7\*7 , it would be computationally impossible to do full depth search. So we limit the depth to which our agent would search the game tree. Once the search reaches depth limit, the agent would treat each node at that depth as the terminal leaf node. At this level, the agent would need to assign a good utility value to each node to help it make a decision on which subtree to pick as its next move. We need to provide a utility function that provides score to each leaf node in a way that the node which has best chance of turning the game in our agent’s favor gets highest utility value. A good utility function would implement a good “[heuristic](https://en.wikipedia.org/wiki/Heuristic)” that corresponds to a human rationale of why a given board should be assigned a specific score. In the Isolation agent I implemented, I tested three heuristics functions and implemented the best one in custom\_score method of game\_agent.py. Here I present the performance of three of my utility functions and rationale for picking the best one to implement in game\_agent.custom\_score.

**Heuristics and their Performance :**

custom\_score : explain heuristic. Describe performance with pic of tournament.

custom\_score\_2 : explain heuristic. Describe performance with pic of tournament.

custom\_score\_3 : explain heuristic. Describe performance with pic of tournament.

**Best Heuristics selection :**