***Heuristic Documentation:***

**Taimoor Ahmad 2019511:**

The heuristic function is called every time it is MAX’s turn and returns the best possible move from the available options. The function itself takes the last node selected by MIN as a parameter. In the heuristic, all children of the last node are taken, and each child is sent individually to the ‘traverse\_dic’ function which traverses the tree from that child as starting point till the leaf node. While doing so, it calculates the distance from the starting point to the leaf node (i.e no of turns it will take to reach leaf node from current state of game). This distance is considered as weight. The weight’s modulus is taken by 2 (weight%2) and if it is equal to 1, then the current node allows a possibility for MAX to win which means that if MAX was to choose this node, he has a chance to win. Both, weight and minWins is returned (as a class object). This process is repeated for each child node. Best possible move is chosen based on several IF conditions in heuristic function.

If it is possible for MAX to win with more than one choice, then the node which has less weight is preferred. If no choice allows MAX to win, max will choose the one with the most weight, since this will give it a winning chance, incase MIN makes the wrong choice.

The best possible choice is stored replaced only if there is a better choice. The heuristic function returns this choice to the max plays function, which plays this option.