B551 – Assignment 2

1. We needed to search the state space for the best possible move for the game based on the valid moves i.e. dropping a pebble in the column or rotate a column if N > 1. So, for this we defined a successor function which gives us all the valid possible moves for a given state, goal function which will tell us of if we have won the game for a state and a evaluation function which will give us a value based on the conditions of a current state.
2. State space: It consists of all the possible states with an array of length N. It should only contain pebble (o), (x) and (.) such that the number of pebbles is <= n(n+3)/2 in any given board.
3. Successor: This function will return all the valid moves for a given state i.e. it will add a pebble of the current player in each column and rotate each column if N > 1.
4. Evaluation function: It returns an integer which is the most favorable to the current player i.e it checks top N rows, columns and 2 diagonals to find if pebble (x) has a strong hold or pebble (o). It also add weight if 2 pebbles of same kind are adjacent in a row or column as [x,x,x,x,o] is better for player with pebble (x) than [x,o,x,o,x]. We also tried to make the board like a circular list this way it would be easy to find out a good value for rotation moves. We tried to make the board as a circular list to make it easier to check the column rotations and the favorable ones to the current player.
5. Goal Function: This function checks if the current player has won the game and returns a Boolean to the caller.
6. Min Function: Initially checks if the opposite player has won the game and then returns a max negative value else returns a value which is the best move for the opposite player i.e the worst possible move for the current player.
7. Max Function: This function also works like a min function except that it plays the best move for the current player.
8. The function always has a cutoff value being passed to return the value when the depth has been reached.
9. The program works similarly to an Iterative deepening search because of the time constraint specified in the problem statement. It starts off with cutoff K = 2 then increases the threshold K += 2.