**Requirements**

a description of how you formulated the search problem, including precisely defining the state

space, the successor function, the edge weights, and any heuristics you designed; (2) a brief description of how

your search algorithm works; (3) a discussion of any problems you faced, any assumptions, simplications,

and/or design decisions you made; and (4) answers to any questions asked below in the assignment.

Problem 1.

The problem is to find the possible moves that don’t make the current player lose the game immediately and selecting a move which makes the other player’s future moves more restrictive. Passing on to the other player a board which is harder for him/her to win. The problem also includes the possibility to play more aggressively if the other player is already in a bad state.

**State space:** Is the possible configuration of the n-k-cho-cho game. Where each player can place each respective black or white marble as per their choice on n by n board by following a legal move. A legal move is defined as when a respective player can place a either a white or black marble on the board where there is an empty space

**Start state:** The start state is an empty board of n by n, consisting of the size (n) specified by the user. It can also be assumed that white player will always make the first move. The board also includes a parameter ‘k’ which is the max quantity in which a marble of a particular color can be placed in a row, column or diagonal.

**Goal state: There are 3 specific goal state**

1. **Player 1 wins (white marbles) – when player 2 losses i.e player 2** completes a row, column or diagonal of size k with black marbles, where k is provided by the user.
2. **Player 2 wins (black marbles) – when player 1 losses i.e player 1** completes a row, column or diagonal of size k with white marbles, where k is provided by the user.
3. **Draw –** when there are no more blank space on the board where either of the player can place any marble and none of the players have placed their respective marbles in k quantity in any row, column or diagonal

**Successor function:** The successor function take the current board looks at all available place (blank places) places a marble in a legal move fashion with reference to the player and returns the board. These returned board are further evaluated by another function and finally a suggestion is made to the user.

**Cost: The cost is basically the cost associated NEED SOME inputs HERE**

**Heuristic function:** The heuristic function is composed of two parts

* Immediate loss function- This function takes the board and calculates the place where the current player should not make a move or place the marble in order to not lose immediately- This part is based on the condition that the current move to be suggested does not meet the terminal or goal state with the current “k”
* Future evaluator of the all available moves after the immediate loss function this parts calculates the move which makes the next moves of the other player not so lucrative.

Assumptions: The white marble player is the user for us and we are recommending the moves for him/her

Problem 2 Tetris

**State space:**

**Start state:**

**Goal state:**

**Successor function:**

**Cost:**

**Heuristic function:**