29/10/24 Lab-6

6. Implement 8-Queens algorithm using A* search & Hill Climbing Algorithm.

A* search

1. Define the initial state
   → start with empty board
   → initialize a empty list

2. Define the Heuristic function (h)
   → For each state, calculate no. of conflicts between queen already placed.

3. Define the cost function
   → this will be no. of queens placed so far.

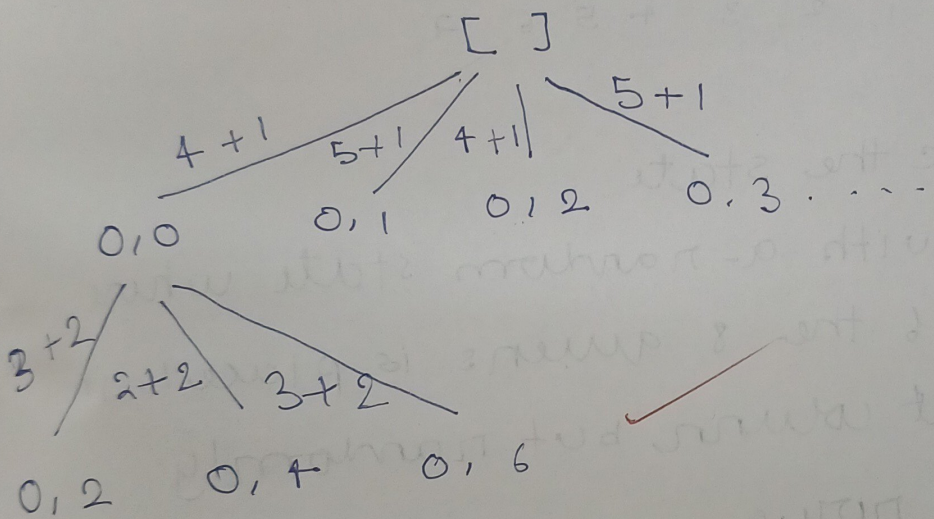
4. define $f(n) = g(n) + h(n)$

5. Expand nodes
   → pop the state with lowest f(n) and place it in the closed list
   → if it has no conflicts (h(n)=0) and all queens are placed, the goal is found

→ otherwise generate all child states by placing queen in next row in each possible col.

6. calculate h(n) for each child state, and add them to open list.

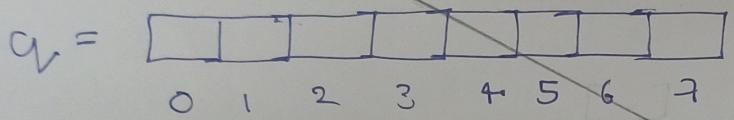
7) Repeat



Hill climbing algorithm

1. Create an array where each index represents a col & the value that represents the row position of queen is in that row.

int [] q = new int[8]



1. Initialize the state
   → start with a random state where each of the 8 queens is placed in different column, but randomly chosen rows.

2. Evaluate the current state
   → h(n) = no. of pairs of queen attacking each other.

3. Generate Neighbours
   → for each column, try moving the queen to every possible row

4. Select the best neighbour

5. select the best neighbour with lowest h(n)

6. move to the new state
   → if the new states h(n) is lower than the current one, move here
   → if no algorithm stops

6. if stuck restart

output (Best soln found)