BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

HYDERABAD CAMPUS

CS F407/EA C461: Artificial Intelligence

Assignment No: 1 Weightage: 8% Date of Submission:18/09/2015

Problem No: 1 (Coding): Given a hop matrix hop[][] and a frog.Frog is sitting outside the matrix. Frog hops to cell index (0,0) and it is counted as one step.Frog can traverse in any one of the four directions at a time,that is East,West,North or South.Each cell of matrix contains a pair (a,b).

a can have a value **0,+1 or -1**.

- 0 indicates that frog can go in any one of four directions.
- +1 indicates that the cell is a bridge that takes frog from one index to another in forward direction(East or South)crossing b-1 cells,that is if frog is at cell (i,j) and it is a bridge then it can go to either (i,j+b) cell or (i+b,j) cell.
- -1 indicates that the cell is a tunnel that takes frog from one index to another in backward direction(West or North) crossing b-1 cells, that is if frog is at cell(i , j) and this cell is a tunnel then it can go to either (i , j-b)cell or (i-b , j) cell.

If a is 0 then b indicates the maximum number of hops frog can take in any one of the four directions and land to another cell, if b is 4 and a is 0 then frog can take 0 or 1 or 2 or 3 hops at a time in any direction. One move is counted as one step.

Crossing a bridge is counted as one step and crossing a tunnel is counted as one step.

Write a program(in any language of your choice) to search for a path from index (0,0) to index(n-1,n-1) of least number of steps using **A* search**.

Devise a heuristic to solve the problem and write a code that gives minimum number of steps required to move from source to destination as output. Fig. 1 shows a sample case with 4*4 hop matrix.

(0, 2)	(+1, 2)	(-1, 1)	(0, 1)
(-1, 1)	(+1,2)	(+1, 1)	(+1, 2)
(0, 2)	(-1, 1)	(+1, 1)	(0, 1)
(-1, 3)	(0,1)	(-1, 2)	Destination

Fig.1

Here two paths of minimum steps are possible.

(0,0)->(0,1)->(0,3)->(1,3)->(3,3)
Or
(0,0)->(2,0)->(2,2)->(0,1)->(3,3)

Therefore minimum 5 steps required.

Problem No: 2 (Coding): Below is a game played on a square board with opposing pieces (black and white) lined up along the end ranks. The players take turn to move, and skipping a move is not allowed. The pieces can move only straight forward, one square at a time (onto unoccupied squares), and can capture an opposing piece only one square diagonally forward (just like regular pawn capture in chess). Figure 2-b shows what the board can look like after a number of moves. The object is to get a pawn to the other side of the board; the game is drawn if the player whose turn is, it cannot move. The given board is 8 * 8. But it works well for any size greater than 3* 3.

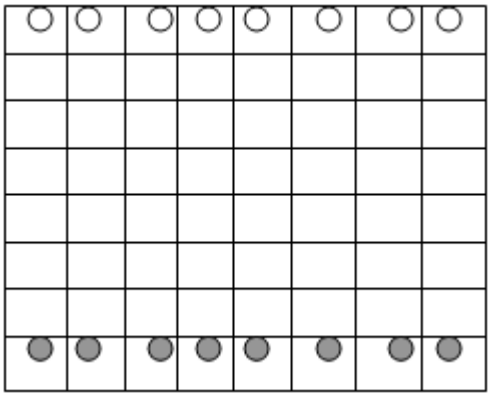

Fig 2-a.

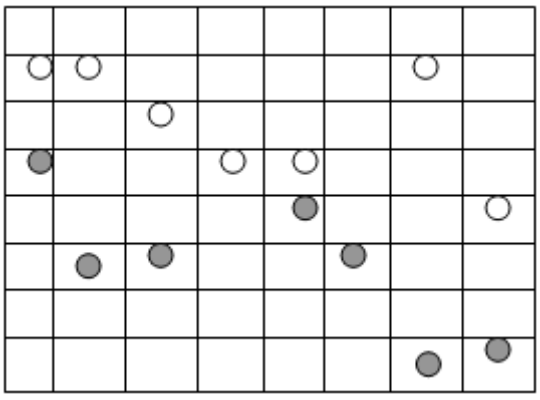

Fig. 2-b

Write a program to play the above game, using **Minimax search with alpha-beta pruning**. Here, you will need to devise a board evaluation function to calculate utilities. One possibility (no doubt not the best) is simply to give both players one point for every square they have moved towards the opposite end of the board (summed over all pieces) and then to subtract black's score from white's. In addition, of course, board positions in which one player has own will have to be assigned huge (positive or negative) utilities. Once you

have decided how to represent a board state and compute its utility, you will need to write a program to generate, for any given board state, all the states that can be reached from it in one move. Then apply the minimaxAB algorithm directly. If you can write a **graphical interface**, it makes the program more fun to play. But, it is optional.

Once you have got the program working, you can investigate the effect of α-β pruning. By counting the number of times that the condition $\alpha > \beta$ is or is not satisfied as the algorithm runs, you can determine the frequency of pruning. By playing the same game with α-β disabled, you can count the number of nodes that were pruned off.

Mode of Submission:Form your own groups of four. Make a compressed folder containing all your submissions and a readme file. Name it by using ID number of anyone from the group and submit it in CMS on or before 18/09/2015.

For any queries related to assignment drop mail to h2104103023@hyderabad.bits-pilani.ac.in