9610

Postlab Exp1

1. What is the easiest trick to win Tic Tac Toe?
A simple trick to win Tic Tac Toe is put your 'X' in any corner, then whatever may be opponents move put another X at the last position of same row or colun or to the diagonal
After the opponents following move place at X at remaining of these empty boxes.
This is a simple trick to win.

2.

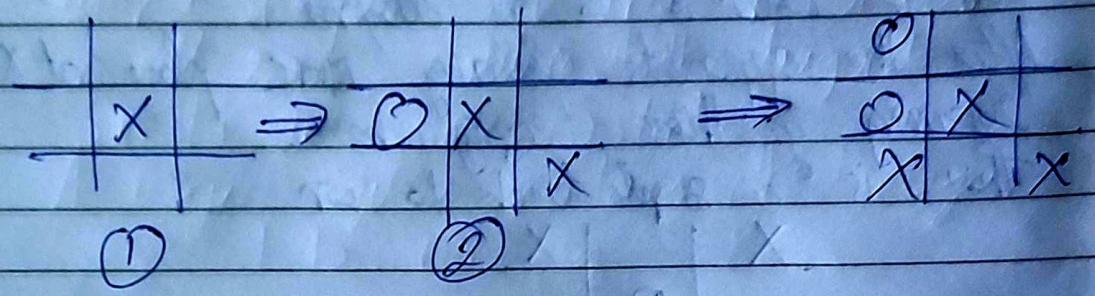
X		X
	O	
O		X

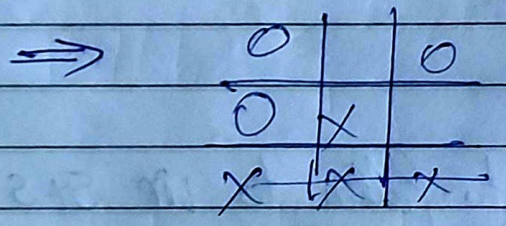
2. What is the algorithm to follow to win 5×5 Tic Tac Toe?
The Advanced Tic Tac Toe strategy of or 5 x 5 Tactical Approach is the algorithm one can follow to win 5x5 Tic Tac Toe.
Het where player should.
1) Aim to occupy center square
2) Place next X in attempt to make winning lines
3) Block Opponent
4) Fork moves
5) Play defense
6) Adjust your strategies be flexible
7) Control the center situation
8) Maintain Pressure play in attack mode.

Teacher's Signature:

Q3 Is there any way to never loose at Tic Tac Toe?
To a strategies which ensure that you never loose a game doesn't ensure you to win it every game

1) If you go first
Place X on centre





Q4. What can Tic Tac Toe help you with?
In learning and understanding concepts of AI Tic tac toe is perfect example to illustrate basic concepts of game theory.
It we can understand min max algorithm and the decision making involved.
We can to understand & practice search algorithms like depth first search, BFS, heuristic; also brute force method.

Teacher's Signature:.....

Parlab Exp 2

Q1 What is the relationship between tic tac toe and magic Square?

A The game of Tic Tac Toe can also be implemented using the concept of magic square of 3x3 magic square.

The Sum of number of all rows column, diagonal is always the same = 15.

2	7	6
9	5	1
4	3	8

Hence player can always compute the score of the board to decide the next move to win.

C: Computer H: Human.

eg.

H		C
	C	

Diff = 15 - (5+6)
= 15 - 11 = 4

The difference should not be greater than 9, hence computer can place C in cell with score 4 and win.

eg 2.

		C
	C	C
H		H

Diff = 15 - 6 - 8
= 8

But it is not empty so compute

Teacher's Signature:

const int n.

Q2. What is magic Square of order n?

A magic Square of order n is an arrangement of n^2 numbers, usually distinct integers such that n numbers in all rows, columns, both diagonals sum to the same constant.
A magic square contains the integers from 1 to n^2.

Postlab Exp 3

Q1 What is the time complexity of the Water Jug Problem?

1) Using BFS method.
Time complexity : $O(n*m)$
Space Complexity : $O(n*m)$

where n and m are the quantity of jug 1 and jug 2.

2) Using DFS method:

As Each state has six possible next states (filling, emptying, pouring each jug so branching factor is 6.

Teacher's Signature:.....

The time complexity is exponential $O(b^d)$ where d is the depth of the search tree.

Postlab Exp 5.

Q1. Explain the time complexity of the A* Algorithm.

The time complexity of A* depends on the quality of the heuristic function.
In a worst case algorithm can be $O(b^d)$ where b is the branching factor i.e average number of edges from each node and d is the number of nodes on the resulting path.

Q2. What are the limitations of A* Algorithm?

A* Algorithm may consume significant memory and processing resources.
A* heavily relies on the quality of the heuristic function.
If heuristic function is poorly designed and does not accurate to estimate the distance to

to the goal the algorithm's performance and optimality may be compromised
A* Algorithm might miss the best path if the guess estimate is overly optimistic
A* Algorithm can be computationally expensive for very complex problems.

Q3. Discuss A*, BFS, DFS & Dijkstra's algorithm in detail with examples

A* Search Algorithm

A* Search combines the advantages of informed and uninformed search by using a heuristic function to guide the search towards the goal state efficiently.
It evaluates nodes based on a combination of the cost to reach them from the start and an estimate of the cost to reach the goal from them.

Breadth First Search

BFS explores all nodes at the current de level before moving to the next level
It guarantees finding the shortest solution path, but it may require a larg amount of memory, especially for deeper

Teacher's Signature:....

Search trees.

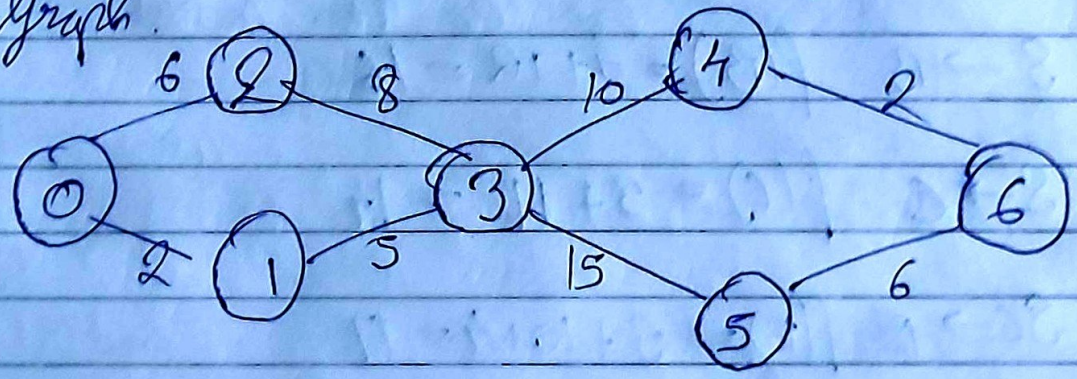
Depth first Search

DFS explores as far as possible along each branch before backtracking. It can be memory efficient, but may not always find the shortest solution, as it can get stuck in deep branches.

Dijkstra's Algorithm

is an algorithm for finding the shortest paths between nodes in a weighted graph. Dijkstra's original algorithm for finding the shortest paths between nodes in a weighted graph, producing a shortest path tree

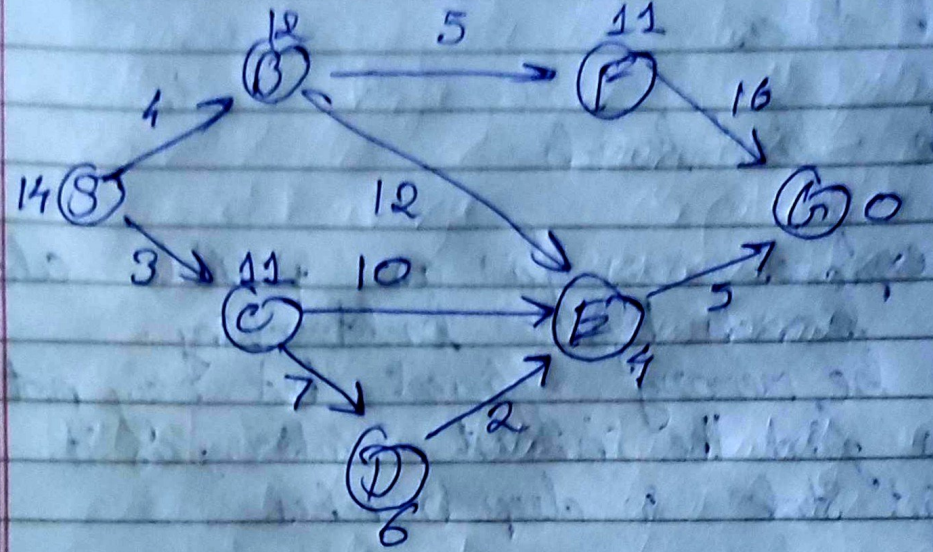
Ex. Graph.



Teacher's Signature:

A*

Ex.



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

$g(n)$ = actual cost for start node to n

$h(n)$ = estimated cost.

$S \to S$ $f(S) = 0 + 14 =$

$S \to B$ $f(B) = 4 + 12 = 16$

$S \to C$ $f(C) = 3 + 11 = 14$

$SC \to E$ $f(E) = 3 + 10 + 4 = 17$

$SC \to D$ $f(D) = 3 + 7 + 6 = 16$.

$SCD \to E$ $f(E) = 3 + 7 + 2 + 4 = 16$

$SCDE \to G$ $f(G) = 3 + 7 + 2 + 5 = 17$

Teacher's Signature:....

BFS

S C B D E F G

DFS

S B F G E C D

BFS

S C B D E F G

DFS

S B F G E C D

Postlab Exp 6.

1). What is the difference between A* and AO* algorithm?

A*	AO*
1. Not designed for handling changes in the environment.	1. Specifically designed to adapt to changes without initiating a new search.
2. Primarily uses AND operation considering one path at time.	2. Uses OR and AND operations exploring multiple plans simultaneously.
3. Explores fewer nodes.	3. May explores more nodes due to adaptability
4. Less suited for high uncertainty or frequent environmental changes.	4. Excels in situations with uncertainty, quickly adjusting response to new information.

Requires a complete restart of search after an environmental change

Eliminates the need for a full restart

Saves time & computational resources when changes occur.

Well suited for static environment with consistent node costs

Particularly in dynamic environment where costs

Q2 Why A0* algorithm only works when heuristic values are underestimated?

A* algorithm has a key property: is admissibility: an admissible heuristic is one that never overestimates the cost to reach a goal.

Overestimating means that the heuristic estimate is higher than the actual final path cost.

If we use an inadmissible heuristic, the algorithm can wind up doing tons of superfluous work examining paths that it should be ignoring, and possibly finding suboptimal paths because of exploring them.

Postlab Exp 7.

Q1. What are the advantages and disadvantages of state space search?

Ans Advantages:

1. State space search systematically explores the possible states and transitions of a problem, ensuring a comprehensive examination of potential solutions.
2. It allows for the representation of complex problems in a structured manner, with states representing configurations and transitions depicting possible actions or moves.
3. State space search can be applied to a wide range of problems in robotics & game playing to natural nlp & scheduling.
4. It can adapt to various problem types including deterministic, stochastic, & adversarial scenarios, making it applicable to diverse set of challenges.
5. Informed decision making can be done using heuristic functions by acquiring domain specific knowledge.

Disadvantages:

Exponential growth: State space can grow exponentially with the size of the problem.
Storage and managing a large state space can require significant memory resources.

Teacher's Signature:

Search process can be time consuming specifically state space is large or if the algorithm does not employ efficient heuristics.
Cannot perform optimally in stochastic or partially observable environments
Problems with a large number of possible actions for each state can lead to a high branching factor.

Q2 What are the advantages and disadvantages of hill climbing approach?
Advantages:
1. The algorithm is straightforward to understand and implement.
2. It's memory efficient, maintaining only the current state data
3. It often converges swiftly to a solution, which is beneficial in scenarios where time is critical

Disadvantages:
1. The algorithm can become stuck at locally optimal solutions that aren't the best overall.
2. Its tendency to focus on the immediate vicinity limits its exploration, potentially overlooking globally optimal solutions

- The quality and effectiveness of the solution found heavily depend on starting point.

Q3. Describe variations of Hill climbing approach?

Ans. a) Simple Hill climbing:

1. Evaluate the initial state. If it is a goal state then stop & return success.
   Else initial state = current state.
2. Do Loop until solution is found:
   or there are no new operators present which can be applied

```
IF    current state = goal state
      // Stop
      return Success
Else If  Score > current state
         currentstate = Score
         Continue
Else
         Continue.
```

b). Steepest Ascent Hill climbing:

```
If    initial state = goal state
      return Success
Else
      in current state = initial state
```

Teacher's Signature:

Loop until Solution state is found
no new operators present which are applicable

If Stunapplied : State → currentstate ⇒ new state

IF current state = goal state
return success
Else if succ > current
best current state = succ
continue

~~Else~~
~~continue~~

exit.
current state = best state

exit

23. Stochastic hill climbing

Evaluate IF initial state == goal state
return success

ELSE
current state = initial state

LOOP
unapplied != state
better state = successor(state) // generate neighbours

IF better state == goal state
return success

Else

current state = better state

Continue

Exit.

Q. Solve the Block world problem by using the STRIPS method.

Eg Lets consider 3 blocks A, B, C

Introducing STRIPS operators.

Pickup (x), : picks block x from table

Putdown (x) puts block x on the table

Unstack (x, y) : Picks up block x which is stacked on y

Stack (x, y) : stacks block x & y

Representation of Pickup (x):

act : Pickup (a)

pre : On Table (a), Clear (a), Handempty

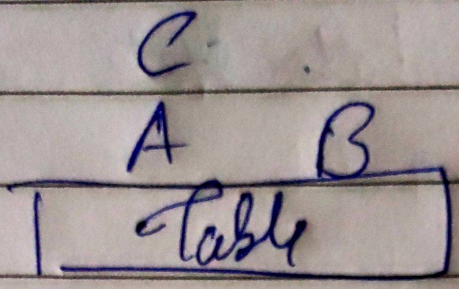
add : Holding (a)

del : On Table (a) Clear (a), Handempty.

initial State:

On (C, A), On Table (A), On Table (B), Clear (C), Clear (B)

Teacher's Signature



On(A,B) ∧ On(B,C)