Victoria University
of Bangladesh

Assessment Topic:

Final Assessment

Course Title: Artificial Intelligence

Course Code: CSI-341

Submitted To:

Md. Shahin Khan Shanto

Co-Ordinator & Lecturer, Department of Computer Science & Engineering

Victoria University of Bangladesh

Submitted By:

Ruhul Amin

ID: 2120180051

Department: CSE

Semester: Summer-2022

Batch: 18th

Submission Date: 6th October, 2022

ID: 2120180051
Page- 1

Answer to the question no-1. a)

Blind Search Algorithm: Blind search algorithm also called uninformed search algorithm. Blind search algorithm do not have any domain knowledge. It works in a brute force manner. It has no ~~knowledge~~ ~~knowleg~~ knowledge about how far the goal node is, it only knows how to ~~travel~~ traverse and distinguish between a leaf node and goal node. It examines every node without any prior knowledge hence called blind search algorithm.

Blind Search Agorithms are mainly three types:

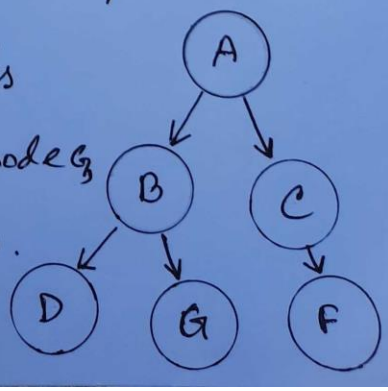
Breadth-first search (BFS)
Depth-first search (DFS)
Uniform cost search.

BFS : In breadth-first search, the tree or the graph is traversed breadthwise, i.e. it starts from a node called search key and the explores all the neighbours nodes of the search key at the depth-first and then moves to the next level nodes. It's queue support first in first out (FIFO).

The time complexity for breadth-first search is b^d where b is average number of child nodes for any given node and is depth.

The disadvantage of this algorithm is that it requires a lot of memory space because it has to store each level of nodes for the next one. It may also check duplicate node.

Example: If the search starts from root node A to reach goal node G, then it will traverse A-B-C-D-G.



ID-2120180051
Page - 3

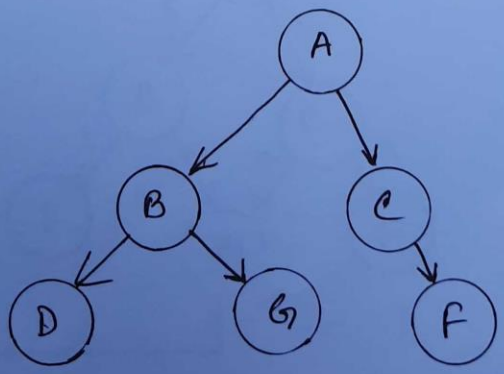
DFS: The tree is traverse depth-wise. It starts from a node called search key and then explores all the nodes. It works on cocept of last in first out (LIFO).

~~The time complexity.~~

It stores nodes linearly hence less space requirement.

The Major disadvantage is that this algorithm may go in a infinite loop.

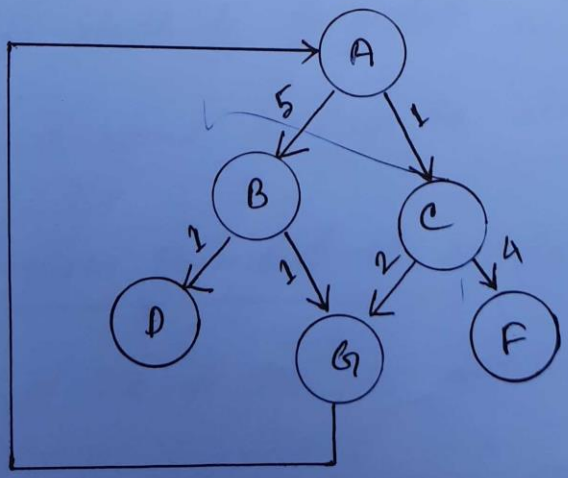
Example: If the search starts from root node A to reach goal node G, then it will traverse A-B-D-G.



Uniform Search: It is different from both BFS and DFS. In this algorithm, the cost comes into the picture. It traverse the path in the increasing order of cost

The time complexity for ~~breath~~ BFS is b^d where b is average number of child node for any given node and d is depth.

Example: If the search starts from the node A to reach goal node G, then it will traverse A-C-G. The cost will be 3.



Answer to the question no 1.(b)

Problem Solving: The problem of AI is directly associated with the nature of humans and their activities. So we need a finite number to solve a problem which makes human easy works.

Four General Steps: These are the following steps which require to solve a problem:

1. Goal Formulation: This one is the first and simple steps in problem solving. It organize finite steps to formulate a target which require some action to achive the goal.

2. Problem Formulation: It decides what action should be taken to achive the formulated goal. In AI this core part is dependent

upon software agent which is consisted of the following components to formulate the associated problem.

component of formulated the associated problem:

Initial State: In this state new methods also initialize problem domain solving by a specific class

Action: This state of problem formulation works with function with a specific class taken from the initial state and all possible actions done in this state.

Transition: This stage of problem formulation integrates the actual ~~cost.~~ action done.

Goal Test: This stage determines that the specified goal ~~achied~~ achieved.

Path costing: This component of problem-solving numerical assigned what will be the cost to achieve the goal. It requires all hardware software and human working cost.

3. Search: Determine the possible sequence of actions that lead to the states of known values and then choosing the best sequence.

4. Execute: Give the solution perform the actions.

—o—

Answer to the question no: 2.a

States: AI problem can be represented as well formed set of possible states. State can be Initial State, Goal State.

Initial States: This state requires an initial state of for the problem which starts AI agent towards a specific goal. In this state new methods also initialize problem domain solving by a specific class.

Action: This stage of problem formulation works with function with a specific class taken from the initial state and all possible actions done in this stage.

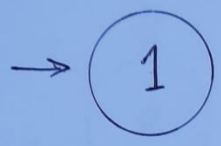
Goal Test: This stage determines that the specified goal achieved by the integrated transition model or not, whenever the goal achieves stop the action and forward into the next stage to determines the cost to achieve the goal.

Path costing: This component of problem-solving numerical assigned what will be the cost to achieve the goal. It requires all hardware software and human working cost.

— o —

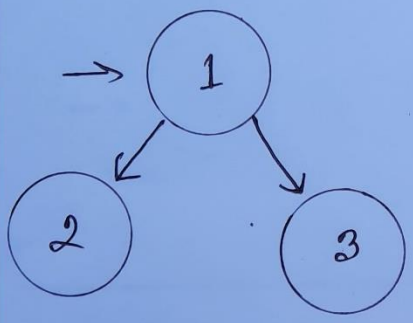
Answer to the question no- 2.b

Breadth First Search: Level order & left to right.



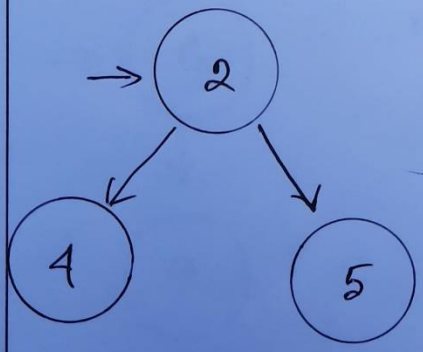
Initial State or starting node.
search neigbour node.

Fringe → 1



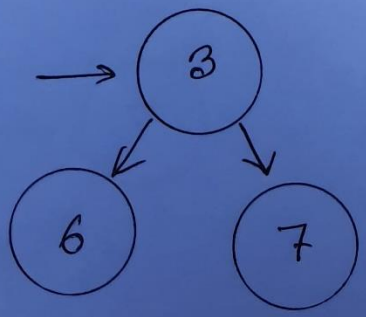
Node 2 and 3 discovered.
Finished ~~too~~ search ~~from~~ from 1.

Fringe → 2, 3



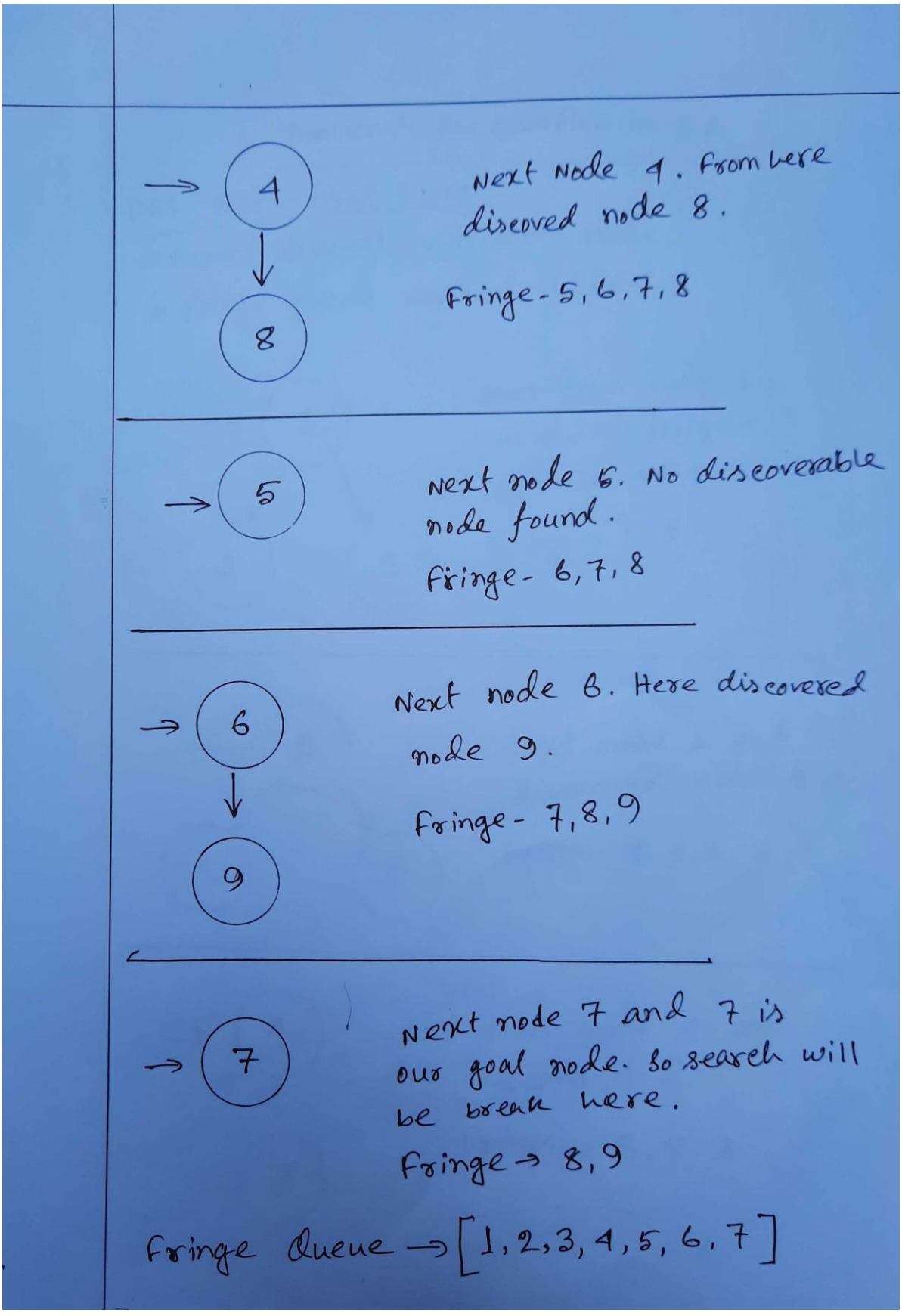
Left node is 2. So from node 2, 4 and 5 dicoved also finished search from 2.

Fringe→ 3, 4, 5



Next. Node- 3. From here 6 and 7 node discovered.

Fringe→ 4, 5, 6, 7


4
8
Next Node 4. From here discoved node 8.
Fringe- 5, 6, 7, 8
5
Next node 5. No discoverable node found.
Fringe- 6, 7, 8
6
9
Next node 6. Here discovered node 9.
Fringe- 7, 8, 9
7
Next node 7 and 7 is our goal node. So search will be break here.
Fringe → 8, 9
Fringe Queue → [1, 2, 3, 4, 5, 6, 7]

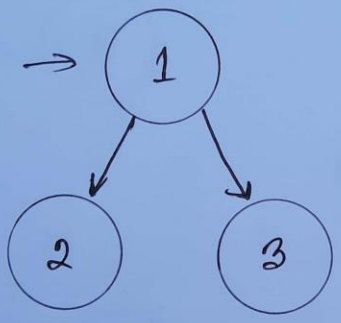
12

Answer to the question no-3 a

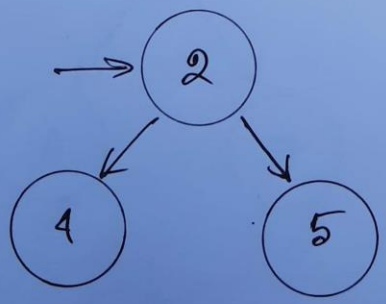
DFS - Depth-first Search:

Expand deepest unexpected node.

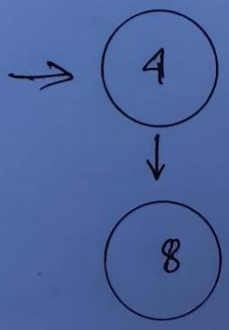
fringe queue method LIFO.



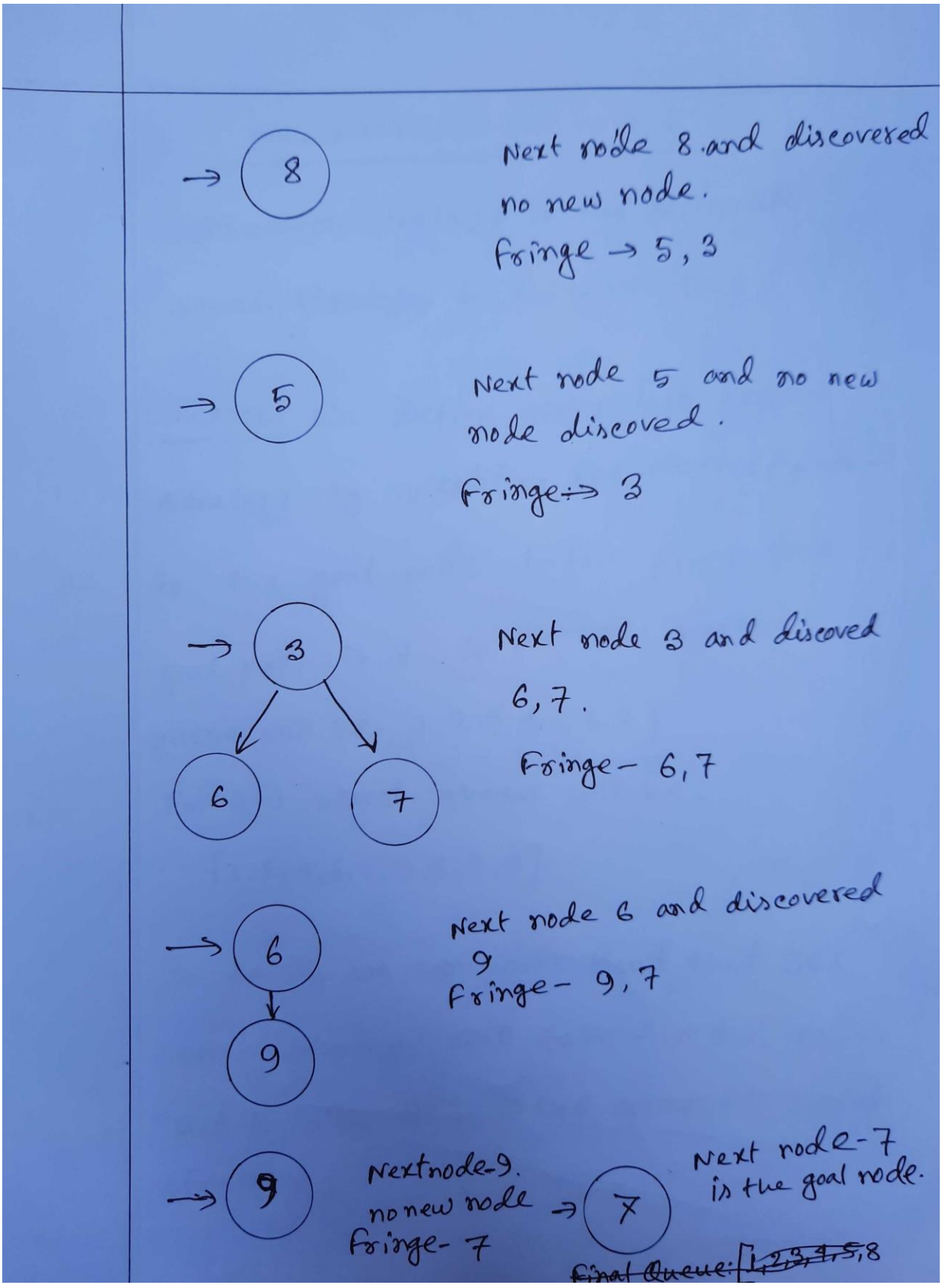
start from node 1.
Initial State fringe - 1
Then discover node 2, 3
Fringe → 2, 3



Next node 2 and discovered 4 and 5.
Fringe - 4, 5, 3



Next node 4 and discovered 8.
Fringe → 8, 5, 3


8
Next node 8 and discovered no new node.
Fringe → 5, 3
5
Next node 5 and no new node discoved.
Fringe → 3
3
6
7
Next node 3 and discoved 6, 7.
Fringe- 6, 7
6
9
Next node 6 and discovered 9
Fringe- 9, 7
9
Nextnode-9. no new node
Fringe- 7
7
Next node-7 is the goal node.
Final Queue: [1,2,3,4,5,8

Answer to the question no 3, b

BFS search strategy is the appropridte search strategy for the given tree.

Reason: We define appropriate search strategy by calculating the shortest path to the goal node. In the given tree goal node is 7. So for BFS search queue will be. [1,2,3,4,5,6,7]

For DFS search queue will be.

[1,2,4,8,5,3,6,9,7]

So, easily we can understand that BFS search strategy will take shortest path. That is why BFS is the appropriate search strategy.

15

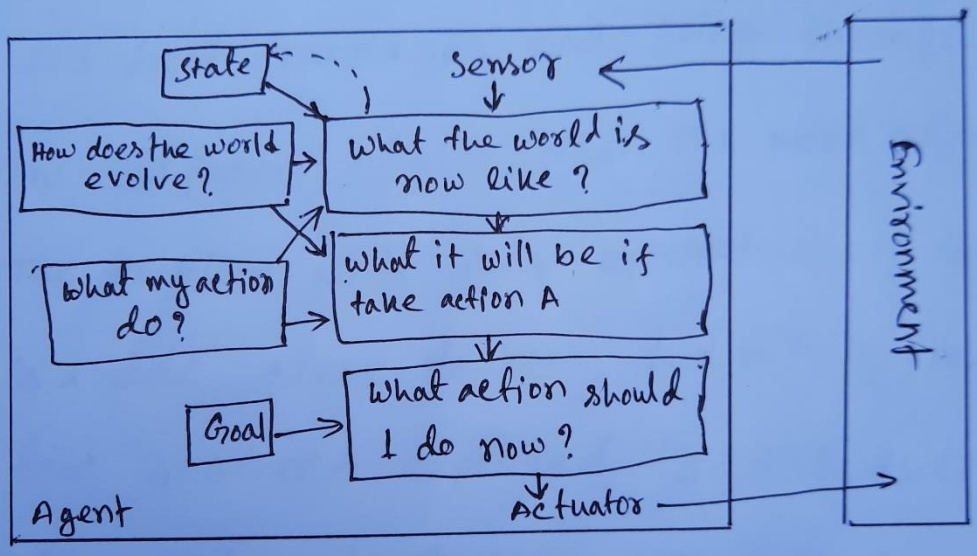
Answer to the question no 4.a

Problem Solving Agent: Problem-solving agent in AI is goal-based agents that focus on goals, is one embodiment of a group of algorithms, and techiniques to solve a well-defined problem in the area of AI. And these agents are different from reflex agents who just have no map state into actions and can't map when storing and learning both are bigger. The different stages that problem-solving agents perform, to arrive at a desired state.

16

Problem Solving Agent

Goal-based agents.



— 6 —

17

Answer to the question no. 4.b

Limitations of DFS strategy:

DFS first traverses node going through one adjacent of root, then the next adjacent. The problem with this approach is, if there is a node close to root, but not in first few subtrees explored by DFS, then DFS reaches that node very late.

DFS may not find the shortest path to a node (in terms of number of edges).

It is possible that may states keep reoccurring. There is no guarantee of finding the goal node.

sometimes the state may also enter into infinite loops.

May find a sub-optimal solution (one that is deeper or more costly than the best solution)

Without a depth bound, one may not find a solution even if one exists.

—o—