2022

Victoria University of Bangladesh

58/11/A, Panthapath, Dhaka, Bangladesh

Artificial Intelligence

CSI-341

Final

Submitted By:

ARS Nuray Alam Parash

ID # 2120180041

18th Batch, BSc in CSE

E-mail: chatokparash@gmail.com

Mobile: 01922339393

Submission Date: 6 October 2022

Submitted To:

Md Shahin Khan (Shanto)

Lecturer

Department of Computer Science & Engineering

Victoria University of Bangladesh (VUB)

Answer to the Question No- 1 (a)

Answer to the Question No- 1 (a)

Blind search Algorithms: Also known as uninformed search, works with no information about the search space, other than to distinguish the goal state from all the others.

Information about search strategies-

(i) **Breadth-first search** – Breadth-first search goes through the tree level by level, visiting all of the nodes on the top level first, then all the nodes on the second level, and so on. This strategy has the benefits of being complete, and optimal as long as the shallowest solution is the best solution. The time complexity of breadth-first search is $O(b^d)$ where b is the branching factor, and, d is the depth of the solution.

(ii) **Depth-first search**- Depth-first search goes through the tree branch by branch, going all the way down to the leaf nodes at the bottom of the tree before trying the next branch over. This strategy requires much less memory than breadth-first search, since it only needs to store a single path from the root of the tree down to the leaf nodes. The time complexity of depth-first search is $O(b^m)$ where b is the branching factor and m is the maximum depth of the tree. Its space complexity is only $b \times m$.

(iii) Depth-limited search- Depth-limited search essentially does a depth-first search with a cutoff at a specified depth limit. When the search hits a node at that depth, it stops going down that branch and moves over to the next to one. This avoids the potential problem with depth-first search of going down one branch indefinitely. The time complexity of depth-first search is $O(b^l)$ where b is the branching factor and l is the depth limit. Its space complexity is only $b \times l$.

(iv) Iterative Deepening search - Iterative deepening does repeated depth-limited searches, starting with a limit of zero and incrementing once each time. As a result, it has the space-saving benefits of depth-first search, but is also complete and optimal, since it will visit all the ~~and~~ nodes on the same level first before continuing on to the ~~and~~ next level in the next round when the depth is incremented. The time complexity of iterative deepening search is $O(b^d)$ where b is the branching factor and d is the depth of the solution. The space complexity is $O(bd)$.

Properties of Search Algorithms - Following are the four essential ~~proper~~ properties of search algorithms to compare the efficiency of these algorithms -

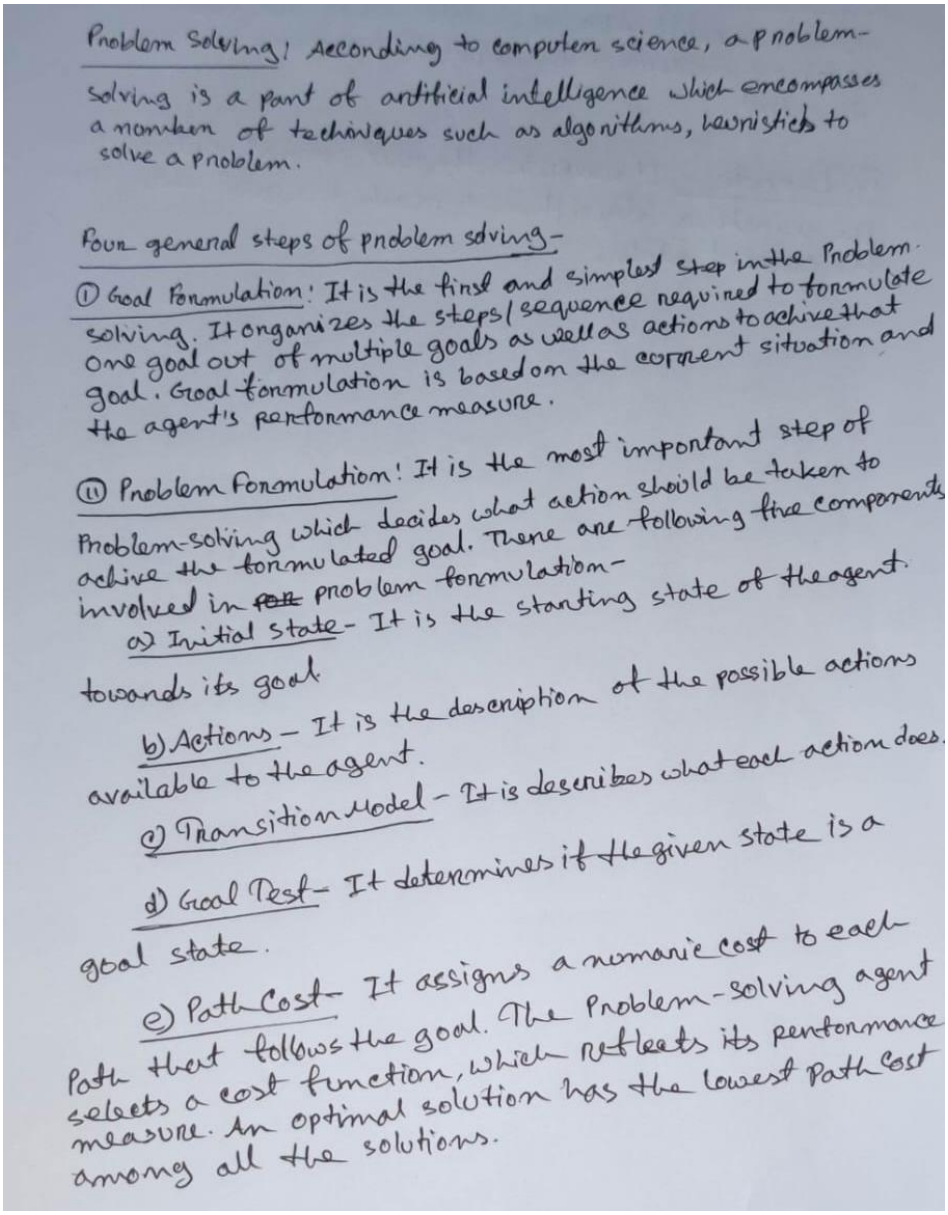
(i) Completeness - A search algorithm is said to be complete if it guarantees to return a solution if at least any solution exists for any random input.

(ii) Optimality - If a solution found for an algorithm is guaranteed to be the best solution (lowest path cost) among all other solutions, then such a solution for is said to be an optimal solution.

(iii) Time complexity - Time complexity is a measure of time for an algorithm to complete its task.

(iv) Space complexity - It is the maximum storage space required at any point during the search, as the complexity of the problem.

Answer to the Question No- 1 (b)



Problem Solving: According to computer science, a problem-solving is a part of artificial intelligence which encompasses a number of techniques such as algorithms, heuristics to solve a problem.

Four general steps of problem solving-

① Goal Formulation: It is the first and simplest step in the Problem-solving. It organizes the steps/sequence required to formulate one goal out of multiple goals as well as actions to achive that goal. Goal formulation is based on the current situation and the agent's performance measure.

② Problem Formulation: It is the most important step of Problem-solving which decides what action should be taken to achive the formulated goal. There are following five components involved in ~~roll~~ problem formulation-

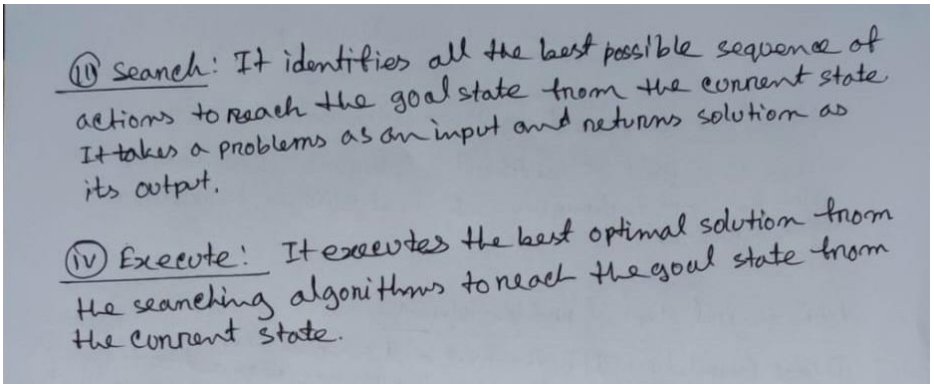
a) Initial state- It is the starting state of the agent towards its goal.

b) Actions – It is the description of the possible actions available to the agent.

c) Transition Model – It is describes what each action does.

d) Goal Test- It determines if the given state is a goal state.

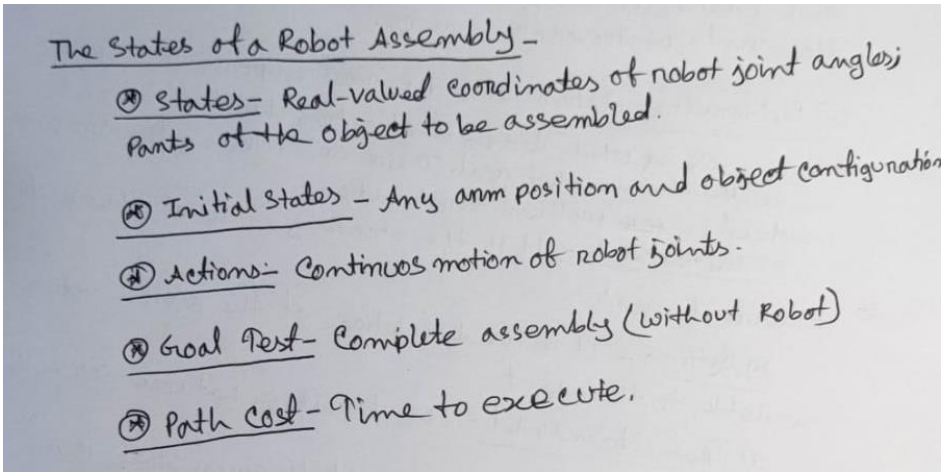
e) Path Cost- It assigns a numaric cost to each path that follows the goal. The Problem-solving agent selects a cost function, which reflects its performance measure. An optimal solution has the lowest path cost among all the solutions.



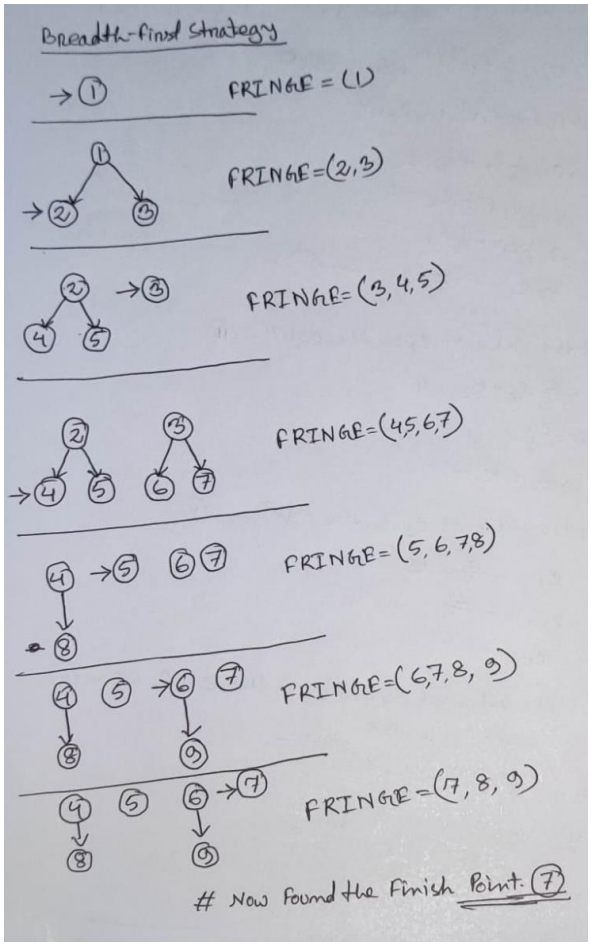
③ Search: It identifies all the best possible sequence of actions to reach the goal state from the current state. It takes a problems as an input and returns solution as its output.

④ Execute: It executes the best optimal solution from the searching algorithms to reach the goal state from the current state.

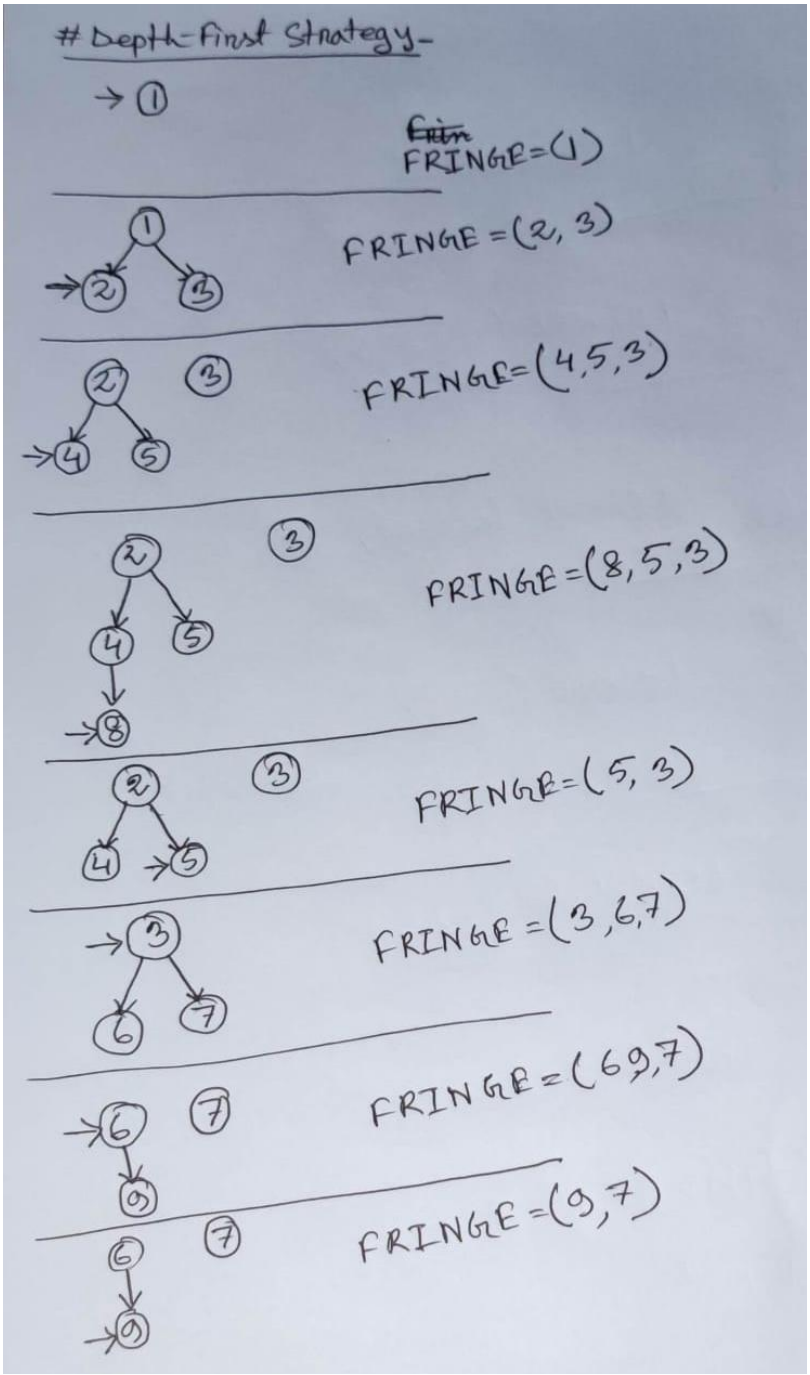
Answer to the Question No- 2 (a)

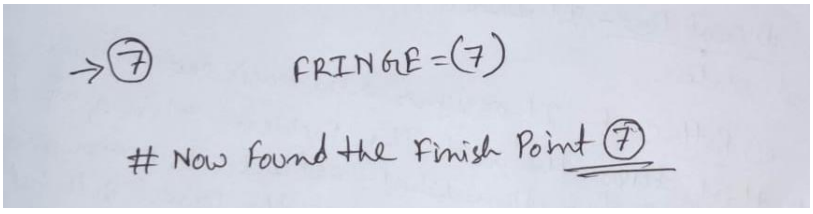


Answer to the Question No- 2 (b)

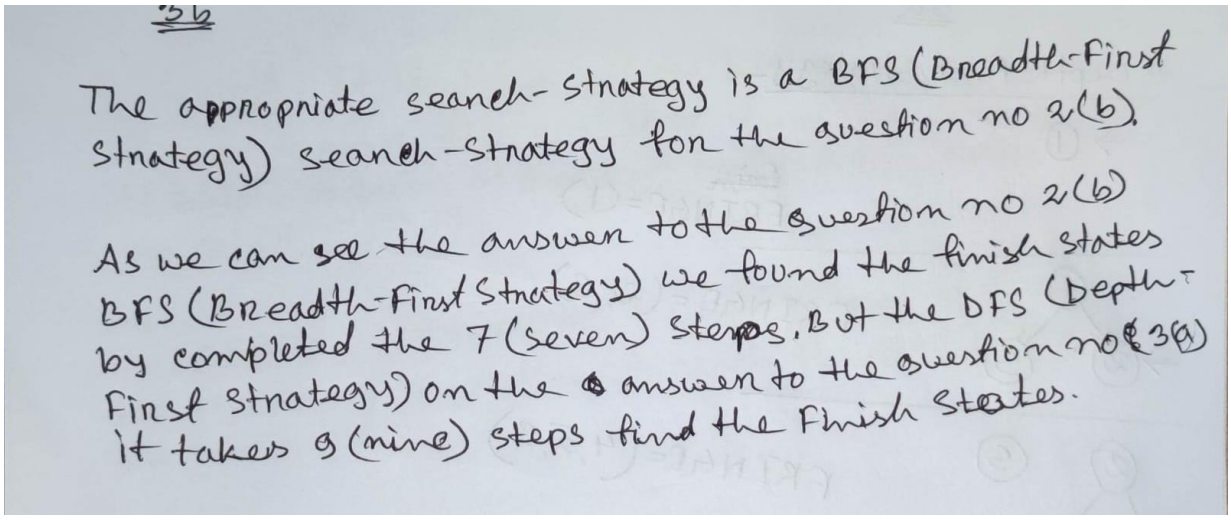


Answer to the Question No- 3 (a)





Answer to the Question No- 3 (b)

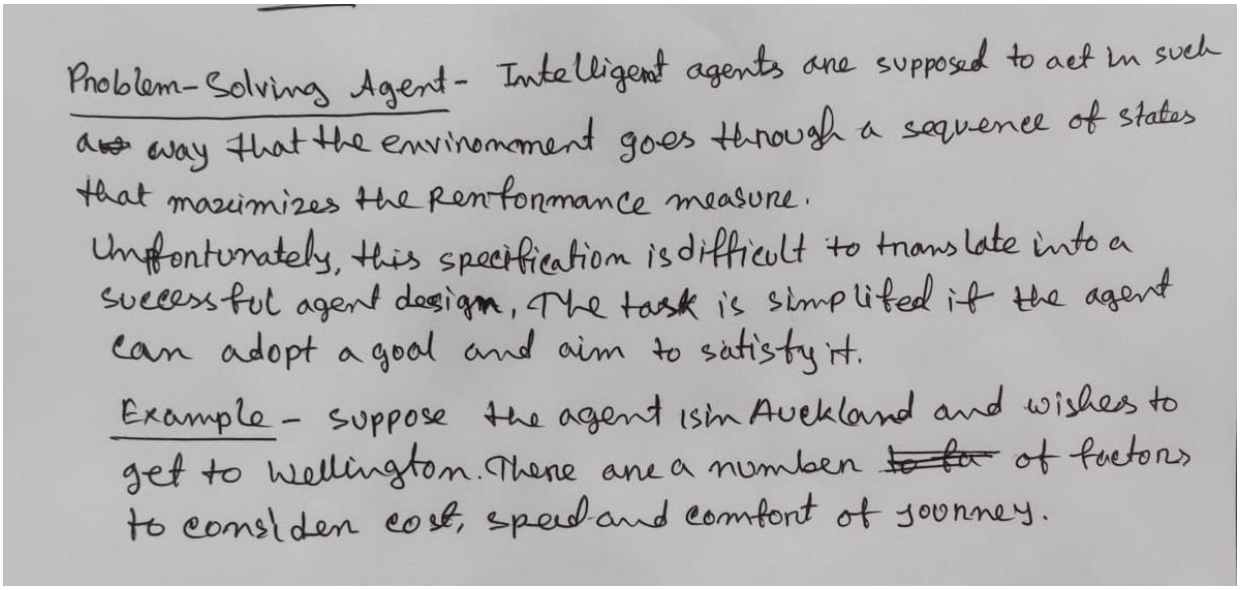


3b

The appropriate search-strategy is a BFS (Breadth-First Strategy) search-strategy for the question no 2(b).

As we can see the answer to the question no 2(b) BFS (Breadth-First Strategy) we found the finish states by completed the 7 (seven) steps. But the DFS (Depth-First Strategy) on the answer to the question no 3(a) it takes 9 (nine) steps find the Finish states.

Answer to the Question No- 4 (a)



Problem-Solving Agent- Intelligent agents are supposed to act in such a way that the environment goes through a sequence of states that maximizes the Performance measure.

Unfortunately, this specification is difficult to translate into a successful agent design. The task is simplified if the agent can adopt a goal and aim to satisfy it.

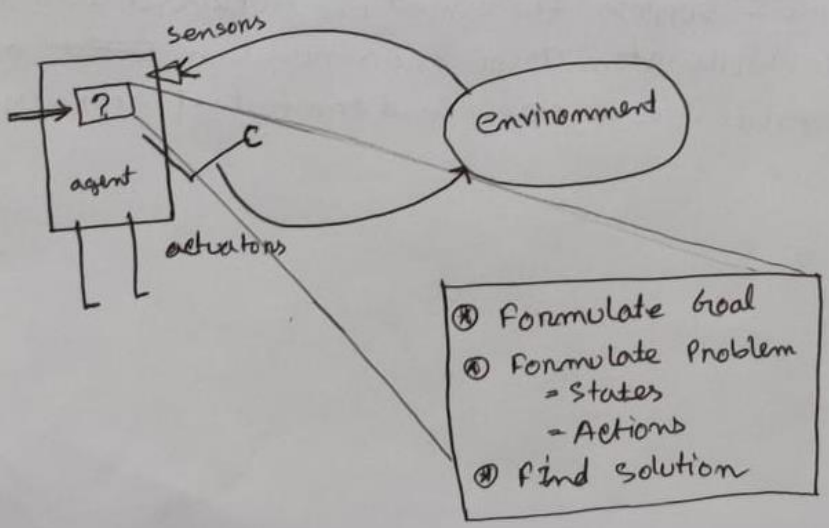
Example - suppose the agent is in Auckland and wishes to get to Wellington. There are a number of factors to consider cost, speed and comfort of journey.

~~tuneat~~

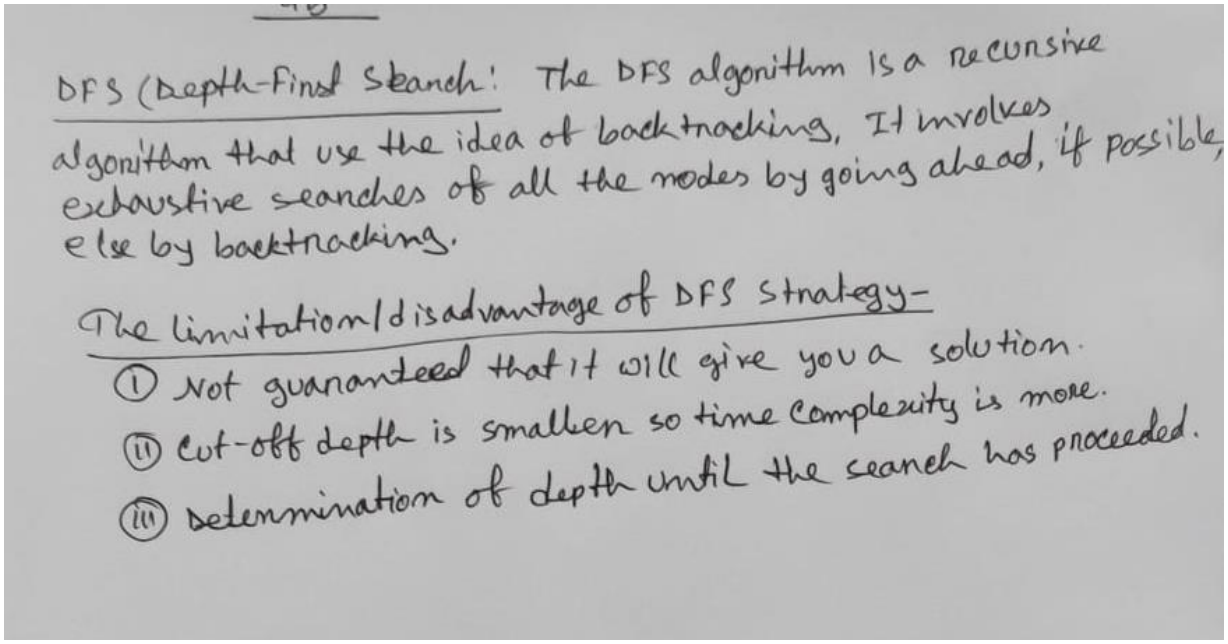
function SIMPLE-PROBLEM-SOLVING-AGENT (percept)
return an action

static: seq, an action sequence
state, some description of the current world state
goal, a goal
problem, a problem formulation

state ← UPDATE-STATE (state, percept)
if seq is empty then
goal ← FORMULATE-GOAL (state)
problem ← FORMULATE-PROBLEM (state, goal)
seq ← SEARCH (problem)
action ← FIRST (seq)
seq ← REST (seq)
return action



Answer to the Question No- 4 (b)



DFS (Depth-First Search): The DFS algorithm is a recursive algorithm that use the idea of backtracking. It involves exhaustive searches of all the nodes by going ahead, if possible, else by backtracking.

The limitation/disadvantage of DFS strategy-

(i) Not guaranteed that it will give you a solution.
(ii) Cut-off depth is smallen so time complexity is more.
(iii) Determination of depth until the search has proceeded.

>> END <<