Assessment On

Course Title: Artificial Intelligence

Course Cord: CSE-341

Department of computer Science & Engineering

Victoria University of Bangladesh (VUB)

Submitted By

Name: Md: Arif Hossain

Reg: 2219150041

Batch: 15th

Program: B.Sc in CSE (EVE)

CSI-341

Ans to the Q. no -1(a)

Blind search, also called uninformed search, works with no information about the search space other than to distinguish the goal state from all the others. The following applets demonstrate four different blind search strategies, using a small binary tree whose nodes contain words. Just enter a word in the text input field and click on the "start search" button in order to begin the search. The nodes will change color to red as they are visited by the search.

* Information about search strategies
  - Breadth - First
  - Depth First
  - Depth Limited
  - Iterative Deepening
  - Javadoc - generated documentation for the applet
  - directory containing the source for the applet

* Breadth First search

Breadth First search goes throught 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 benefit of being complete (if there a solution it will be found) and optimal as long as the shallowest solution is the best soketion.

* Depth - First search

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 node. However, it is potentially incomplete, since it will keep going on down one branch until it finds a dead - end, and it is no optimal if

there's a solution at the fourth level on the first branch tried, and a solution at the second level in the next one over, the solution at the fourth level will be returned.

* 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 the depth, it stops going down that branch and moves over to the next one. This avoids the potential problem with depth-first search of going down one branch indefinitely.

* Iterative Deepening Search

Iterative ~~deeping~~ 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 the also complete and optimal, since

it will visit all the nodes on the same level first before continuing on to the next level in the next round when the depth is incremented.

Ans to the Q. No-1(b)

In artificial Intelligence, search techniques are universal problem solving methods. Problem solving agents in AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result. Problem solving agents are the goal based agents and use atomic representation In this topic, we will learn various Problem solving search algorithms.

Step-01 ⇒ Obtain a description of the problem

This step is much more difficult than it appears. In the following discussion, the word client refers to someone who wants to find a solution to a problem and the word developer refers to someone

who finds a way to solve the problem. The developer must create an algorithm that will solve the clients problem.

step-02 ⇒ Analyze the problem

The purpose of this step is to determine both the starting and ending points for solving the problem. This process is analogous to a mathematician determining what is given and what must be proven. A good problem description makes it easier a perform this step.

step-03 ⇒ Develop a high level algorithm

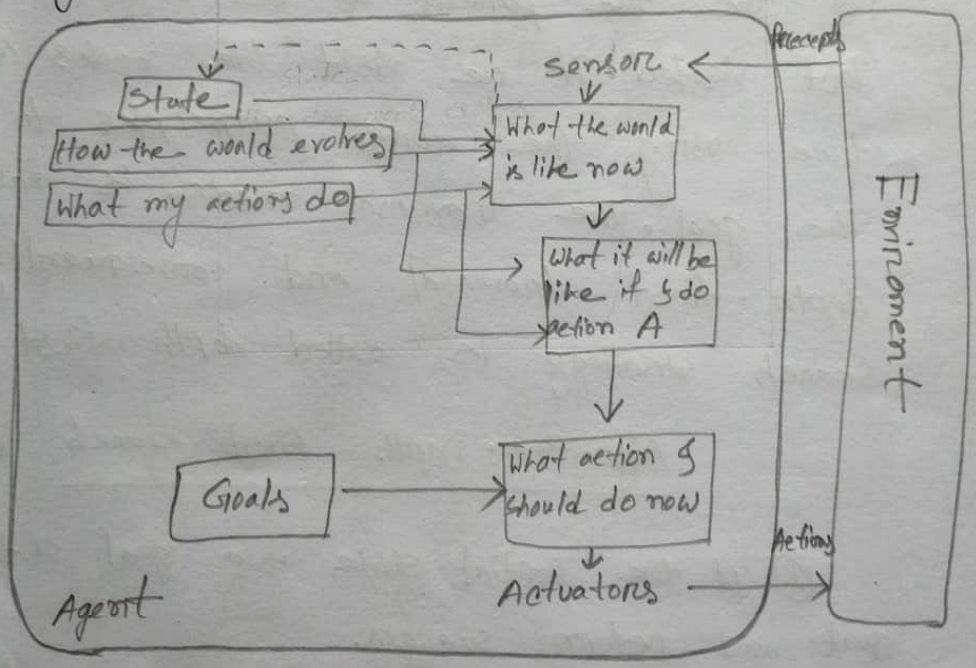
An algorithm is a plan for solving a problem. but plans come in several level of detail. It's usually better to start with a high-level algorithm that inclus the major part of a solution, but leaves the details until later. we can use an everyday example to demonstrate a high-level algorithm.

step-04 ⇒ Refine the algorithm by adding more detail

A high level algorithm shows the major steps that need to be followed to slove a problem. Now we need to add details to these steps but how much detail should we add? Unfortunately the answer to this question depends on the situation. we have to consider who is going to implement the algorithm and how much that person already knows how to do. If someone is going to purchase marks birthday card on my behalf, my instructions have to be adopted to whether on not that person is familiar with the stores in the community and how well the purchaser known my brothers taste in greeting cards.

Ans to the Q. No-4(a)

Problem solving Agent. An agent that tries to come up with a sequence of actions that will bring the environment into a desired state. Search The process of looking for such a sequence involing a systematic exploration of alternative actions.



Ans to the Q.No-4(b)

Breadth first search is an uniformed search technique. We may sometimes search the goal along with the largest depth of the tree, and move up only when further traversal along the depth is not possible we then attempt to find alternative offspring of the the parent of the node (state) last visited. If we visit the nodes of a tree using the above prinei ples to search the goal, the traversal made is called depth-first traversal and consequently, the search strategy is called depth first search.

Algorithm: Depth - First search

① If the initial state is a goal state, quit and return success.

② Otherwise, do the following until success or failure is signaled:

(a) Generate a succesor, E, of the initial state.

(b) Call depth - first search with E as the initial state.

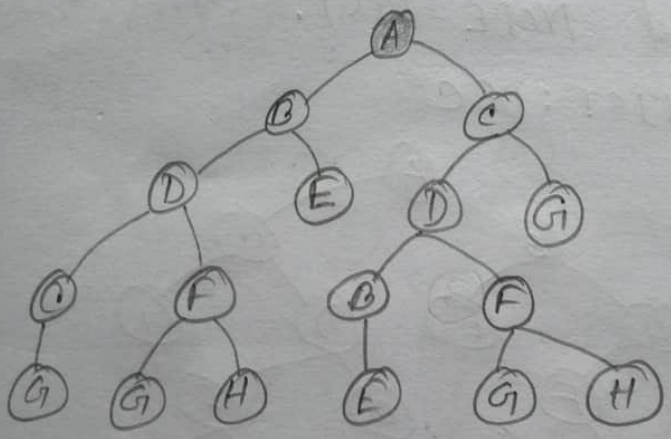
(a) If success is returned, signal success Otherwise continue in this loop.

Depth - First search - Example

let us consider the following tree. Here node A is the source or start or initial node and node G is the goal node.

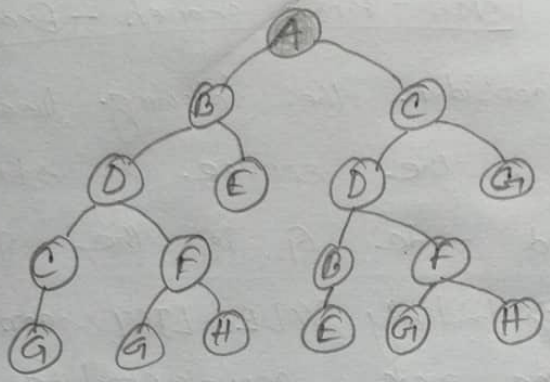
Step 01 ⇒ Initially NODE - LIST contains only one node corresponding to the source state A.

NODE - LIST : A



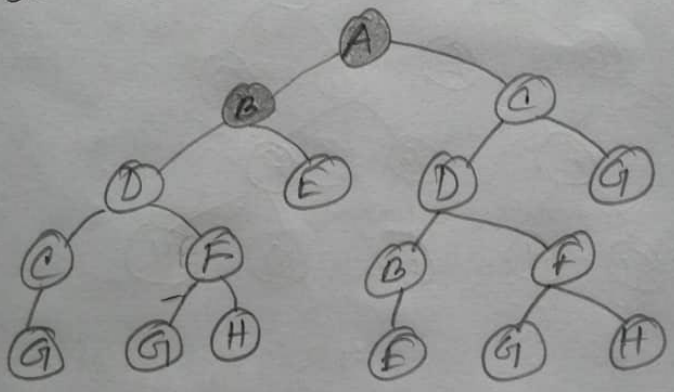
Step: 2 ⇒ Node A is removed from NODE-LIST. A is expanded and its children B and C are put in front of NODE-LIST.

NODE - LIST: BC



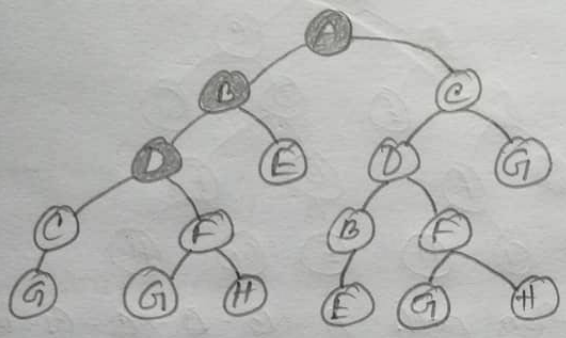
Step: 3 ⇒ Node B is removed from NODE-LIST and its children D and E are pushed in front of NODE - LIST.

NODE - LIST : DEC



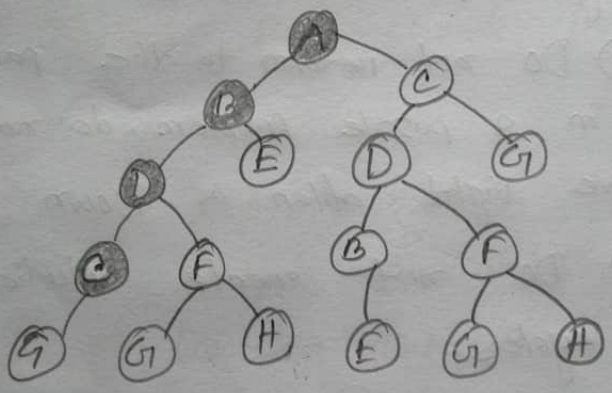
Step : 4 ⇒ Node D is removed from NODE-LIST. C and F are pushed in front of NODE-LIST.

NODE-LIST : C F E C



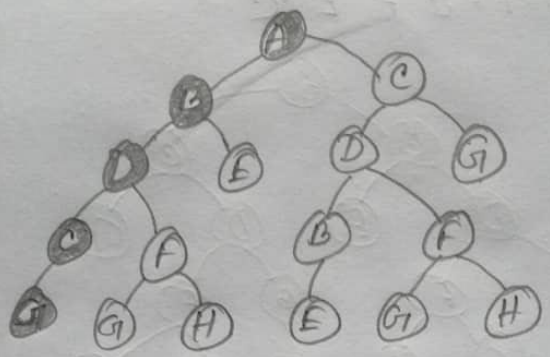
Step : 5 ⇒ Node C is removed from NODE-LIST Its child G is pushed in front of NODE LIST.

NODE-LIST : G F E C



Step 6 ⇒ Node G is expanded and found to be a goal node.

NODE - LIST: G F E C



The solution path A - B - D - C - G is returned and the algorithm terminates.

Ans to the Q. No - 5 (a)

Avoiding Repeated States ⇒

① Do not return to the parent state. (e.g in a puzzle problem, do not allow the up move right after a Down move)

② Do not create solution paths with cycles.

③ Do not generate any repeated states (need to store and check a

Potentially large number of states)

Ans to the Q. No - 5 (b)

A search algorithm is the step-by-step procedure used to locate specific data among a collection of data. It is considered a fundamental procedure in computing. In computer science, when searching for data, the difference between a fast application and a slower one often lies in the use of the proper search algorithm.

Typical Real word Problem

Transaction
* ATM Machine
* Web Application
# Decision Making
* Forecasting

control Problem

* Traffic controller

Seaneching problem

* Seaneh engines

sonting problem

* Transport schedule