victoria university of Bangladesh

student Name	: Md. sohel Rana
student ID	: 2119170011
course title	: Artificial Intelligence
course code	: CSI-341
program	: B.sc in (CSE (Reg)
semesten	: summen-2022
Batch	: 17th

Ans to the QU:NO : 1

(1) (a)

List the blind search algorithms and write their properties :

Thus uninformed search algorithms are also called blind search algorithms. The search algorithm produces the search tree without using any domain knowledge, which is a brute force in nature. They don't have any background information on how to approach the goal or whatsoever.

properties of search algorithms :-

* completeness. A search algorithm is said to be complete when it gives a solution or returns any solution for a given random input.
* optimality
* time complexity
* space complexity
* Breadth-first search
* Depth-first search
* Depth-limited search
* Iterative deepening depth-first search.

properties of search algorithms :-

completeness :-

A search algorithm is said to be complete when it ~~is~~ gives a solution or returns any solution for a given random input.

optimality :-

If a solution found is best (lowest path cost) among all the solutions identified, then that solution is said to be an optimal one.

time complexity :-

The time taken by an algorithm to complete its task is called time complexity. If the algorithm completes a task in a lesser amount of time, then it is an efficient one.

space complexity :-

It is the maximum storage or memory taken by the algorithm at any time while searching.

Breadth-first search

Depth-first search

Depth-limited search

Iterative deepening depth-first search.

(1) (b)

Below four general steps of problem solving :-

- Goal formulation :
  * what are the successful world states.
- problem formulation :-
  * what actions and states to consider given the goal
- search :-
  * determine the possible sequence of actions that lead to the states of known values and then choosing the best sequence.
- Execute :-
  * Give the solution perform the actions.

sometimes, it is not enough to just cope with the problems - they need to be solved. Most people engage in problem solving every day. It occurs automatically for many of the small decisions that need to be made on a daily basis.

For example, when making a decision about whether to get up now or sleep in for an extra 10 minutes, the possible choices and the relative risks and benefits of obeying the alarm clock or sleeping later come automatically to mind.

Larger problems are addressed in a similar way. For example: "I have takes that need to be done by the end of the ~~work~~ week.

Ans to the qu:No: 2

(2) (a)

The states (states, Initial state, Actions, Goal, test, path cost) of a robot assembly:

#states: Real-valued coordinates of robot joint angles; parts of the object to be assembled.

#Initial state: Any arm ~~possible~~ position and object configuration.

Actions: continuous motion of robot joints.

#Goal test: complete assembly (without robot)

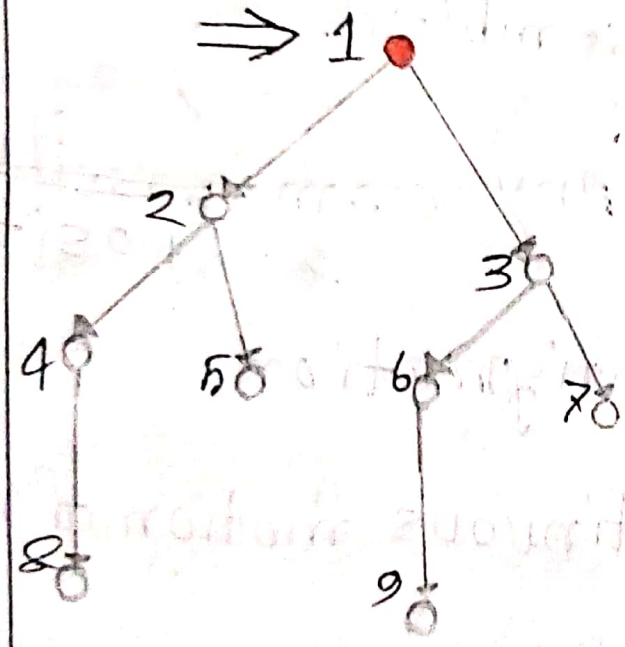
#path cost: time to execute.

(2)(b)

The given tree using BFS search - strategy :-

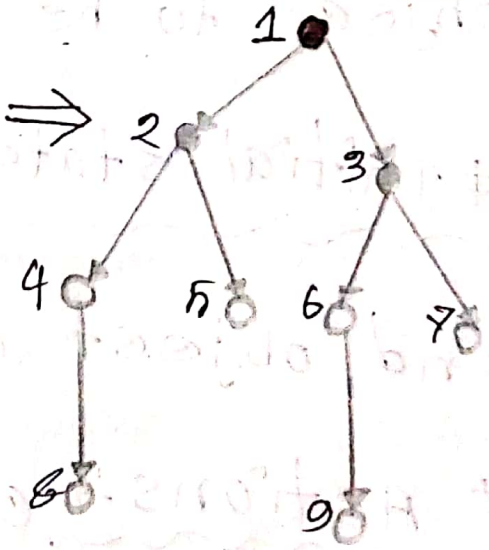
* Implementation : fringe is a FIFO queue
* New nodes are inserted at the end of the queue.

step-1



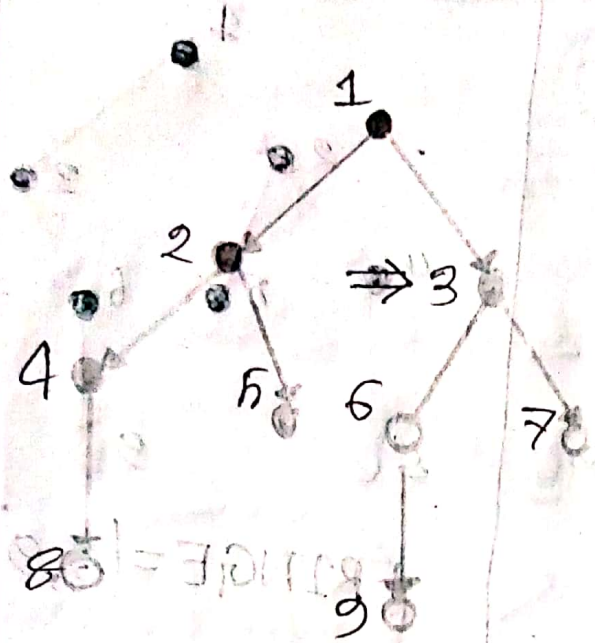
FRINGE = (1)

step-2



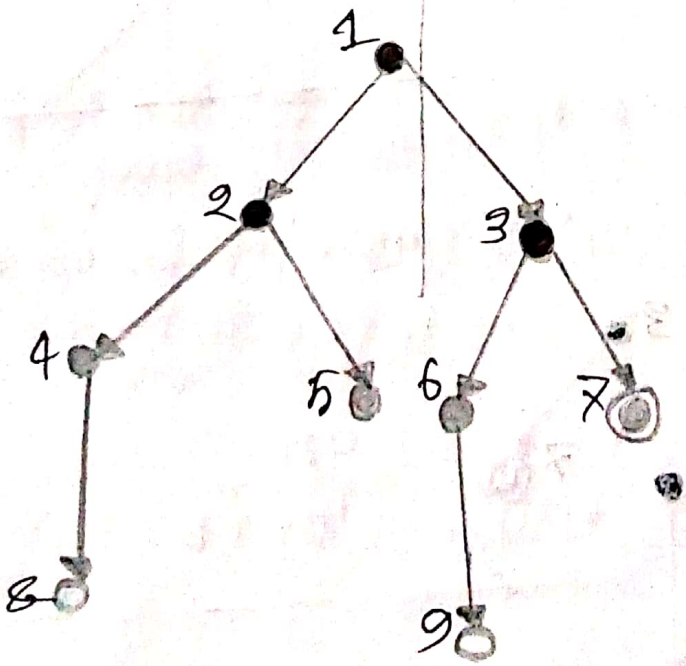
FRINGE = (2,3)

step-3



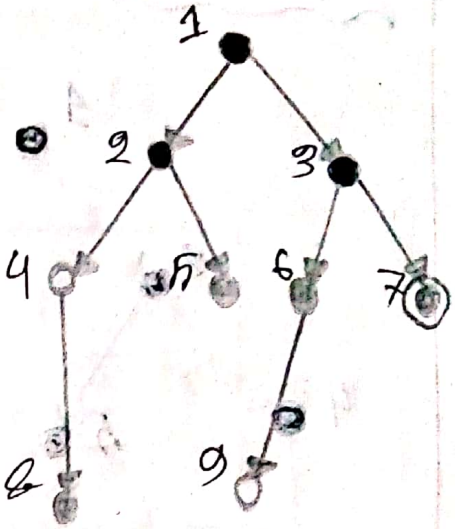
FRINGE = (3, 4, 5)

step-4



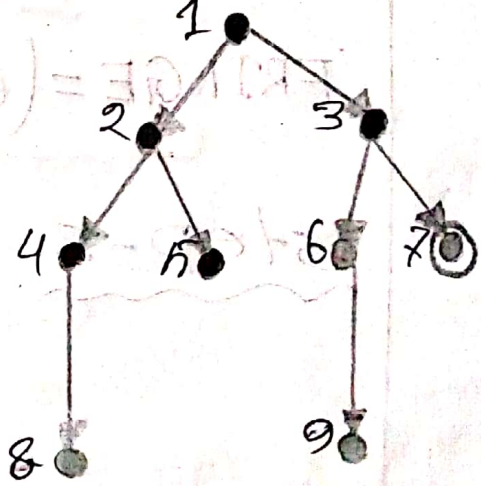
FRINGE = (4, 5, 6, 7)

step-5



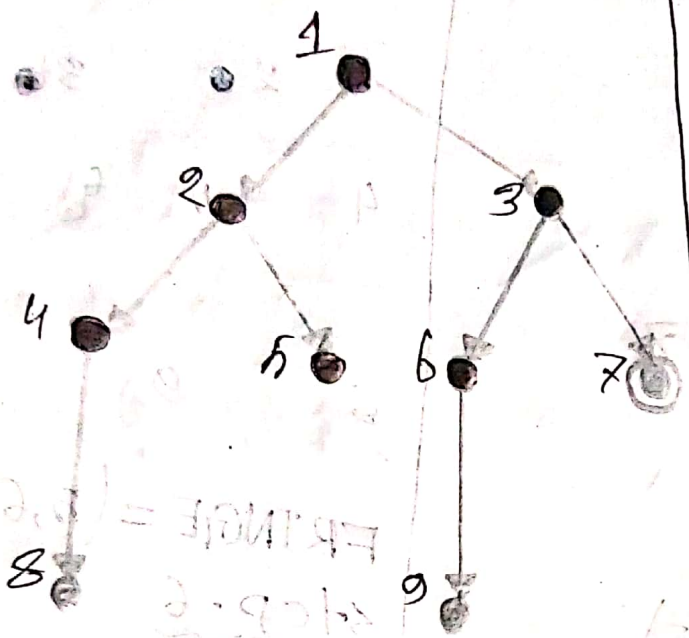
FRINGE = (5, 6, 7, 8)

step-6



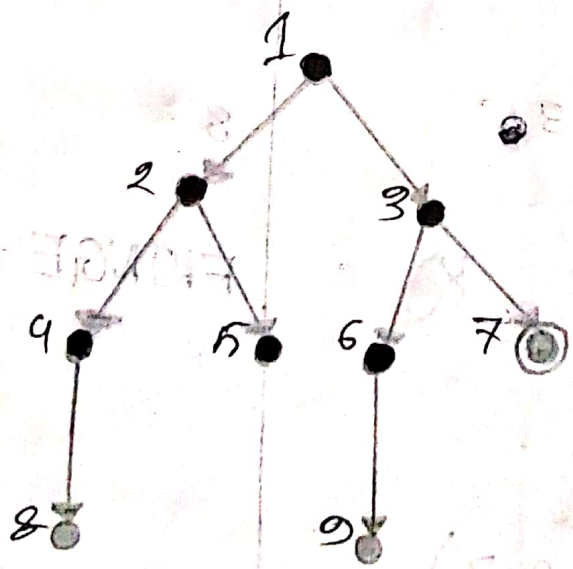
FRINGE = (6, 7, 8)

step-7



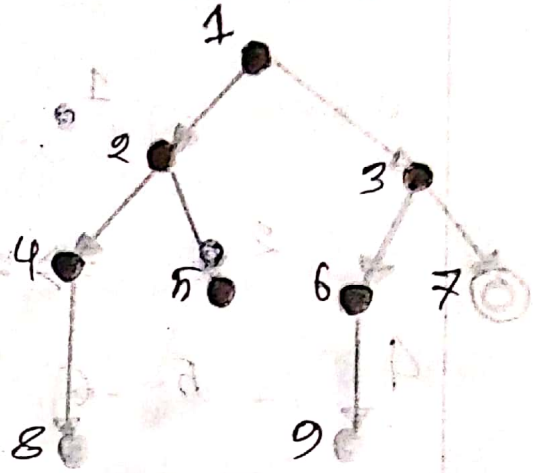
FRINGE = (6, 7, 8)

step-8



FRINGE = (7, 8, 9)

step-9

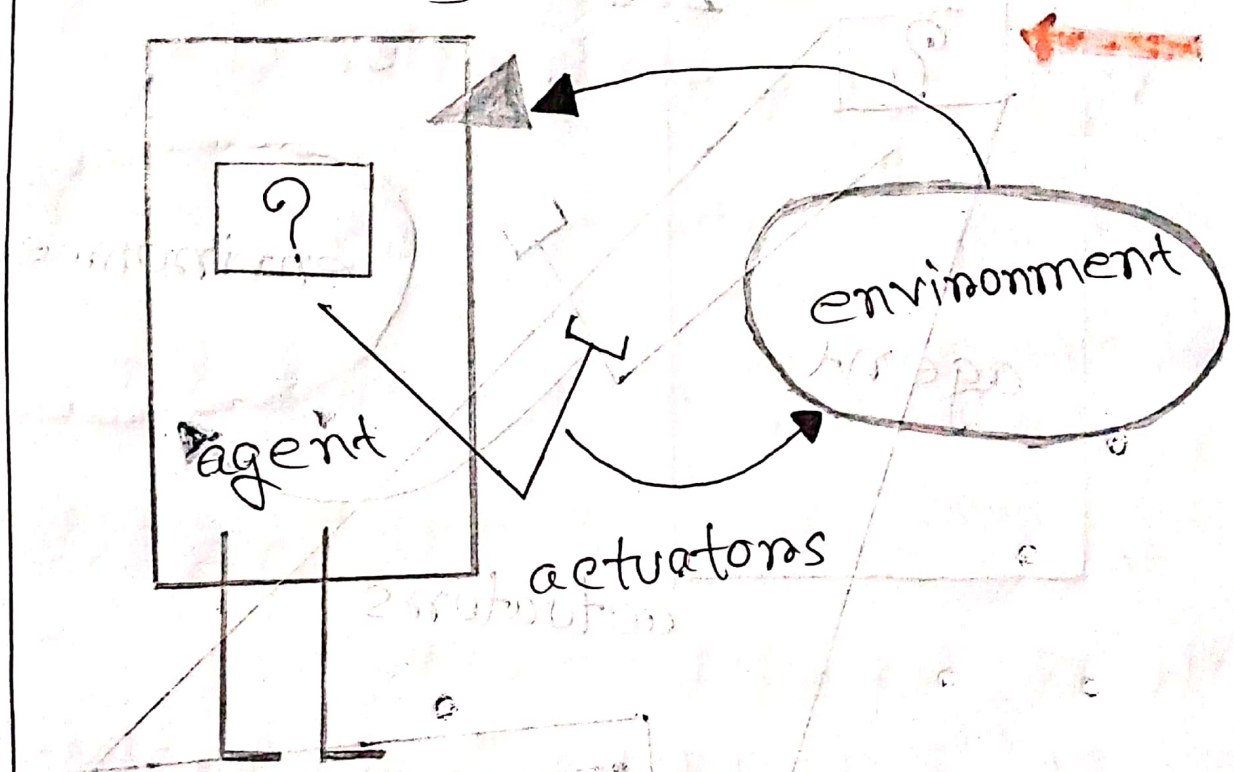


FRINGE = (8, 9)

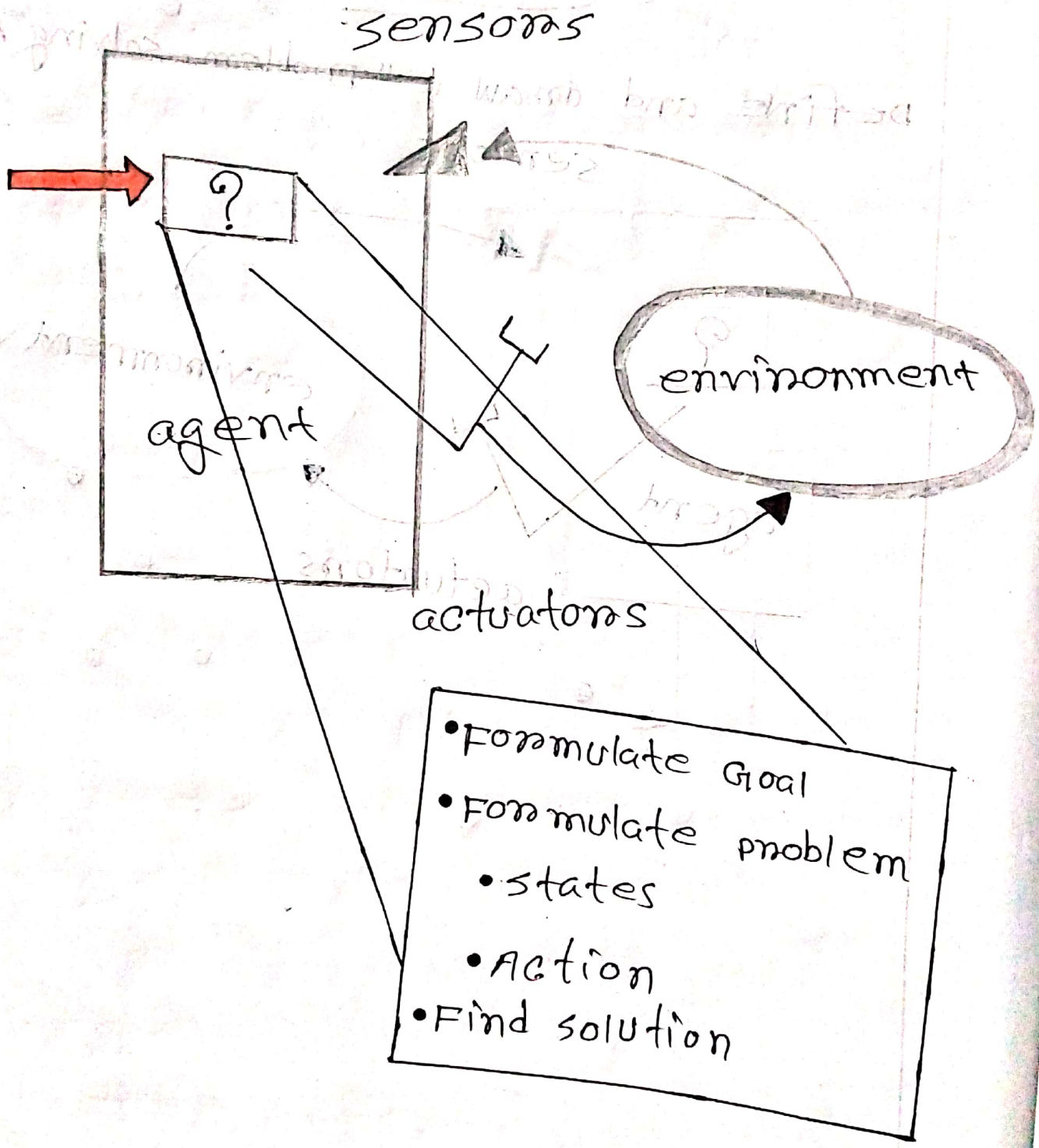
Ans to the Qu.No: 4

(4) @

Define and draw a "problem-solving Agent".



problem-solving Agent



problem-solving Agents

- Goal formulation :- Based on the current situation and the agent's performance measure, is the first step in problem solving.
- problem formulation :- The process of deciding what actions and states to consider, given a goal.
- The process of looking for a sequence of actions that reaches the goal is search.
- Execution :- once a solution is found, the actions it recommended can be carried out.
- "formulate, search, execute" design

problem-solving Agent

```
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
```

④ (b)

The limitation of DFS strategy :-

Depth-first with depth cutoff K (maximal depth below which nodes are not expanded).

Three possible outcomes :-

- solution
- Failure (no solution)
- cutoff (no solution within cutoff).

solves the infinite-path problem.

If K < d then incompleteness results.

If K > d then not optimal.

Time complexity : $O(b^K)$

Space complexity : $O(bK)$

Ans to the QU. NO : 5

(5) (a)

The solution for avoiding repeated states :

requires comparing state descriptions

Breadth-first strategy :

- Keep track of all generated states
- If the state of a new node already exists, then discard the node.

Avoiding repeated states :

Depth-first strategy :

* solution 1 :

- Keep track of all states associated with nodes in current tree.
- If the state of a new node already exists, then discard the node

→ Avoid loops

solution 2 :

- keep track of all states generated so far
- If the state of a new node has already been generated, then discard the node.

→ space complexity of breadth-first

(ii) (b)

Real-world problems

Route finding
touring problems
VLSI layout
Robot navigation
Automatic assembly sequencing
Drug design
Internet searching

Searching Algorithms

Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored. Based on the type of search operation, these algorithms are generally classified into two categories:

1. Sequential search: In this, the list or array is traversed sequentially and every element is checked. For Example: Linear search.

2. Interval search: These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear search as they repeatedly target the center of the search structure and divide the space