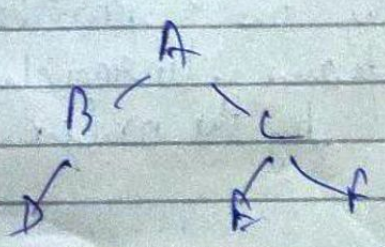
Name- Vishkarsh Sharma
Class- 6CSB
Roll No- 105
RTU ROO- 16EJCCS01

AI Assignment-1

Ans1 BFS → Breadth first search is a vertex based technique for finding a shortest path in graph. It uses a queue data structure which follows first in first out. In BFS one vertex is selected at a time when it is visited & marked then its adjacent are visited & stored in the queue.

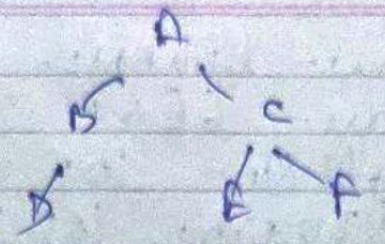


→ Output → A, B, C, D, E, F

DFS - stands for depth first search is an edge based technique. It uses the stack data structure & performs two stages, first visited vertices are pushed into stack & second if there is no vertices then visited vertices are popped.

PAGE NO.
DATE: / /20

Ans-1



⇒ A, B, D, C, E, F.

Ans-2 Informed search algo have info on the goal state which helps in more efficient searching. This info. is obtained by a func that estimates how close a state is to the goal state.

Example → Greedy search & graph search.

Uniformed search algo have no additional info on the goal node other than the one provided in the problem def. The plans to reach the goal state differ only by the order & length of actions.

Ans-3 Constraint satisfaction problem

In AI, constraint satisfaction is the process of finding a solution to a set of constraints that impose conditions that the variable must satisfy. A sol is therefore a set of values for the variable that satisfies all constraints - that is, a point in the feasible region.

The technique used in constraint satisfaction depend on the kind of constraints being considered.

Solving → a constraint problems on finite domains are typically solved using a form of search.

Ans 4→

A* algo	AO* algo
• often known as OR graph algo.	• known as AND-OR algo.
• used to find just one promising path of sol^n.	• used to find more than one promising path or sol^n.
• Application →	• Application
⇒ mostly used in practices	⇒ rarely used in practices application.
Eg → ~~Traffic navigation system.~~	
Eg → Traffic navigation system.	• Eg → alpha-beta pruning for game tree.

PAGE NO.
DATE: / /20

dnt → The great advantage of IDDFS is found in game tree searching where the IDDFS search operation tries to improve the depth def^{n}, heuristics, & scores of searching nodes so as to enable efficiency in the search algo.

- IDDFS gives us hope to find the sol^{n} if it exists in the tree.
- When the sol^{n} are found at the lower depths say 'n', then the algo proves to be efficient in time.
- space & time complexities are expressed as $O(d)$ & here d is defined as goal depth.
- Though the work is done here is more yet the performance of IDDFS is better than single BFS & DFS operating exclusively.
- Quick responsiveness. The early results indications are a plus point in this algo.