Assignment

N. prasanth

22BCE20385.

1. **Missionaries and cannibals problem:**

In the Game there are 3C & 3M in the right side & They want cross a river to reach left side which travelling in a boat. no. of cannibals & no. of missionaries at the time only two travel.

Left (final)   Right (initial)

$$M = 3, C = 3 \quad \& \quad n(Boat) = 2$$

Table:-

S.NO	Final stage (C,M)	Initial stage (C,M)	Boat direction	Action.
1.	(0,0)	(3,3)	—	Initial stage
2.	(2,0)	(1,3)	←	Take 2c to right
3.	(1,0)	(2,3)	→	send 1c to ~~left~~ right
4.	(3,0)	(0,3)	←	send 2c to left
5.	(2,0)	(1,3)	→	send 1c to right
6.	(2,2)	(1,1)	←	send 1c & 1m to left
7.	(1,1)	(2,2)	→	send 2m to left
8.	(1,3)	(2,0)	←	send 1c & 1m to right
9.	(0,3)	(3,0)	→	send 2m to Right
10	(2,3)	(1,0)	←	send 2c to left
11.	(1,3)	(2,0)	→	send 1c to right
12.	(3,3)	(0,0)	←	send 2c to left.
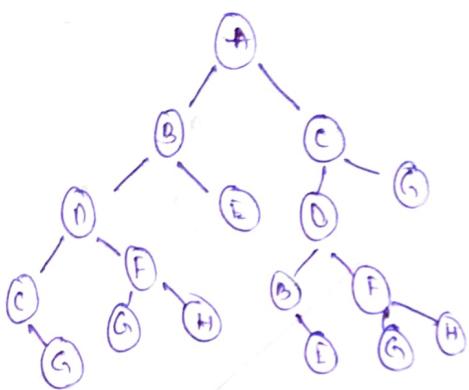
This is goal state.

By following the above path 3C & 3M cross the river

2. for the given tree find the goal state & using BFS & DFS search strategies & Justify which is the best for Given tree.

Given tree:

Algorithm for BFS (Breadth First search)

1. Create a variable called NODE - LIst in to the initial state.

2. until a goal state is found (or) NODE - LIst is Empty

a) Remove the 1st element from NODE - LIst

if NODE - LIst was empty new state them quit.

b) for element E do the following.

i) Apply the rule to generate a new state.

ii) If the new state is a goal state. quit and return then state
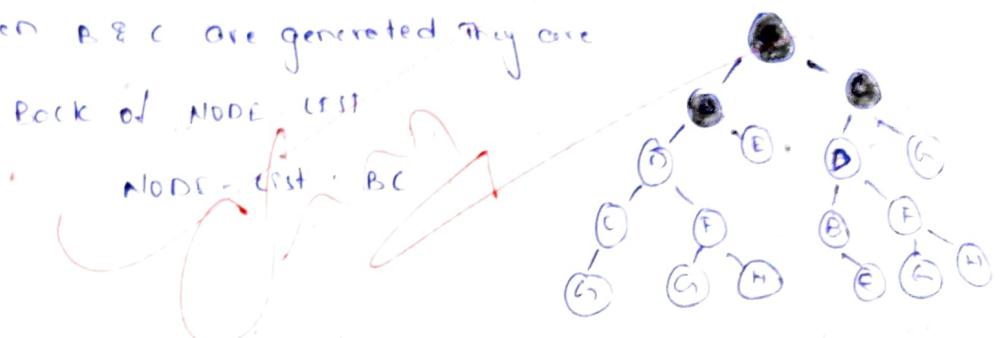
iii) other wise , add the new state to the End of NODE - LIst.

**Step-1** Initially NODE-List contains only one node corresponding the source state A.
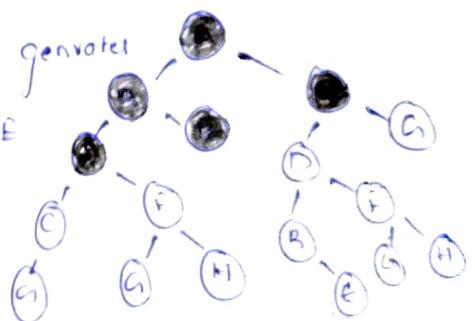
Node - List : A

Step-2: A is removed from NODE list the node is expanded, and it's children B & C are generated. They are placed at back of NODE List

NODE - List : BC
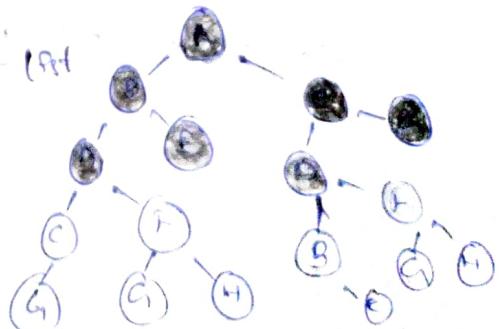
Step-3: NODE B is removed from NODE-List & is expanded. its children, D, E are generated and put at the back of NODE-List : CDE

Step-4: NODE C is removed from NODE-List & is expanded. it's children D & G are added to the back of NODE List

NODE List = D E D G.
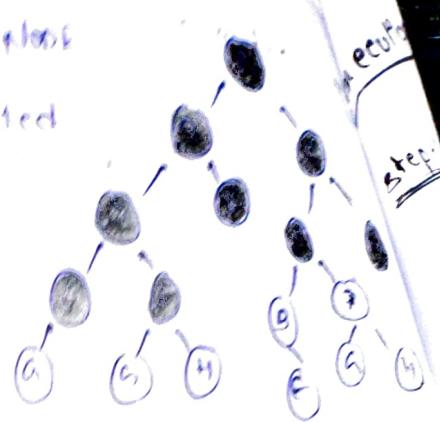
Step-5: Node 'N' to be removed from open

(or) it's children G & H are generated

in the back of node . list

NODE - list - E H G C H
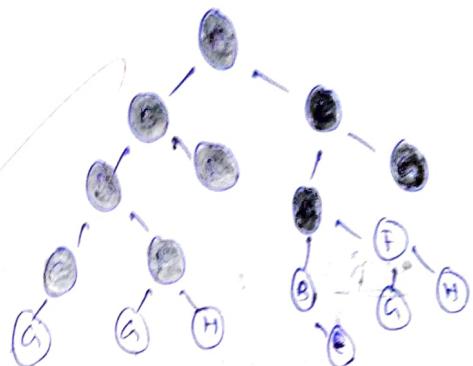
Step-6: Node 'H' is removed from

NODE . list it has no children

NODE list   D H C H

Step-7:

'D' is expanded B & F put in open NODE

list . G C F B F.

**Step-8:** 'G' is selected for expansion it is found to be goal

node .
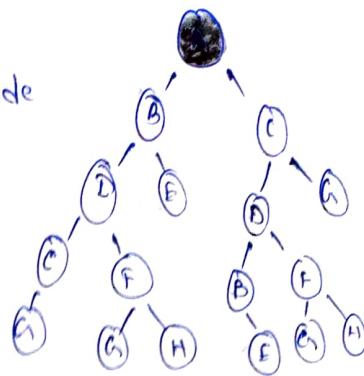
→ Hence the algorithm returns the path A - C - G by the

following the parent out pointers . of the node corresponding an

to G .

algorithm for DFS (Depth first search).

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

2. otherwise do the following until success (or) failure is

signaled.

a) Generate a success, E of the initial state.

if there are no more success, signal failure

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

c) If success is overturned, signal success otherwise

continues in the loop.

**Step-1:** Initially NODE-List contains one node corresponding to the Source state A.
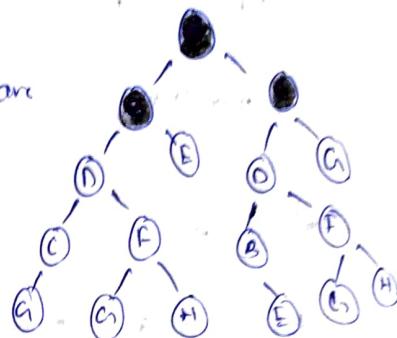
NODE - List : A.

**step-2:**
A is Removed from NODE-List A is expanded and its children B and C are front of NODE - List
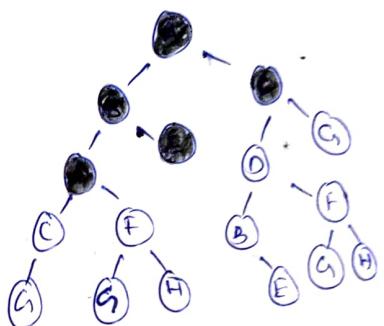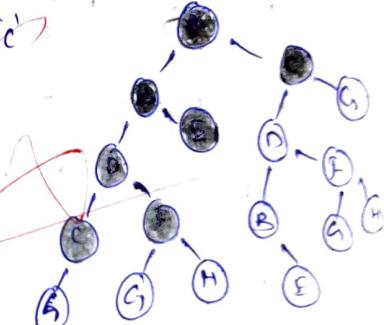
NODE-list : BC

step-3: NODE-B is removed from NODE list & its children 'D' & 'E' are pushed in front of NODE-List

NODE-List :- DEC
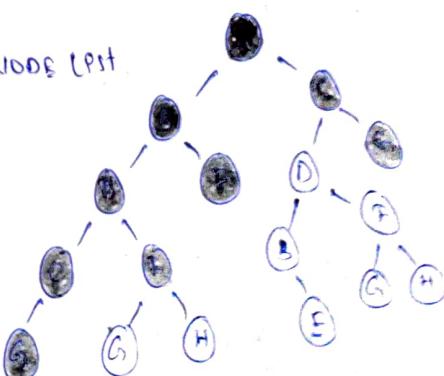
**step-4:** NODE 'D' is removed from NODE List 'C' 'E' 'F' are pushed front of NODE List

NODE-List : EFEC
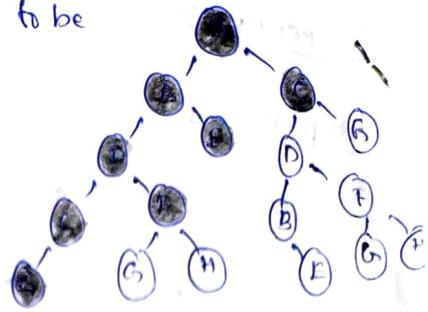
**step-5:-** NODE 'C' is removed from NODE -List child 'a' is punished in front of NODE List

NODE-List :- GFEC

<u>Step-6 :</u> NODE 'a' is expanded & found. to be a goal node.

NODE LIST : GFEGC
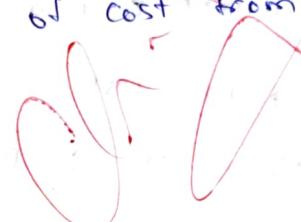
3. Exploain A star & Ao-star Algorithm A-star Algorithm:- $f(n) = g(n) + h(n)$.

→ A-star is a heuristic search algorithm, that aims to find the shortest path b/w two modes in a graph.

→ It contains the cost to reach a mode (known by $g(n)$) & a estimate of cost from the mode to the goal ($h(n)$).

Ao - star Algorithm.

→ Ao star is a variant of the A-star algorithm designed to be more flexible & adaptive in changing Environment.

→ It belongs to family of informed search algorithm utilizing heuristic (or) Estimates cost functions to guide the search process

→ Ao * can repair its path & adapt to dynamic situation by considering a known base, where the A star & goal state are predefined.