Name - Palak Chaturvedi

Roll no - 2K20/MC/90

Assignment - 1

AI MC 312

1) Time Complexity of DFS :

$$N_{DFS} = [(d+1) + (b^{d-1} - 1)/(b-1)]/2$$

$$= [b^{d-1} - bd + b - d - 2)/2(b-1)$$

$$= b^d/2 \quad \text{for large } d$$

Time Complexity of BFS

$$N_{BFS} = (b^d - 1)/(b-1) + (1 - b^d)/2$$

$$= (b^{d-1} + b^d + b - 3)/2(b-1)$$

$$\approx b^d(b-1)/2b \quad \text{for large } d.$$

$$N_{BFS}/N_{DFS} = (b+1)/b$$

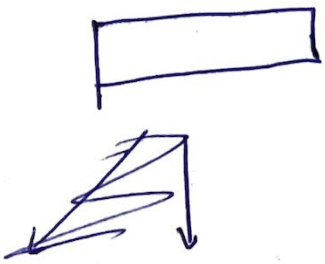
Space complexity of DFS : ~~linear~~

$$O_{DFS} = (b-1)(d-1) + b$$

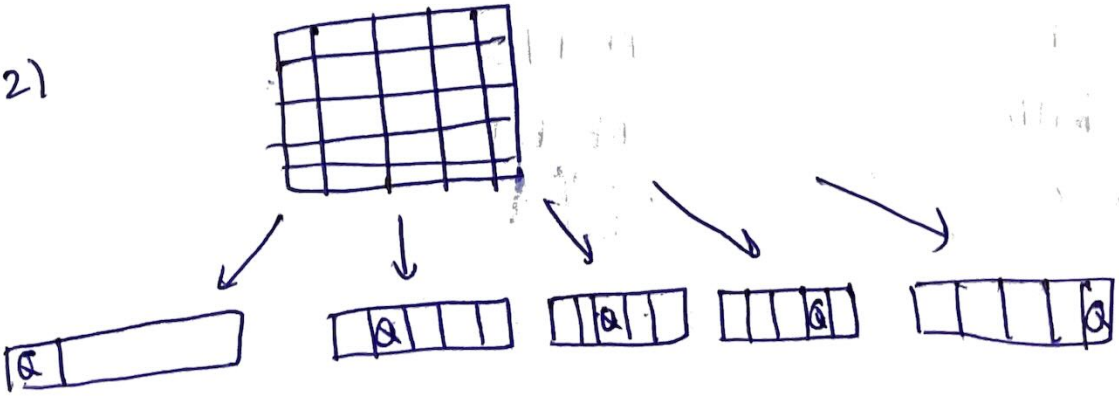
$$= d(b-1) + 1$$

Space complexity of BFS : exponential

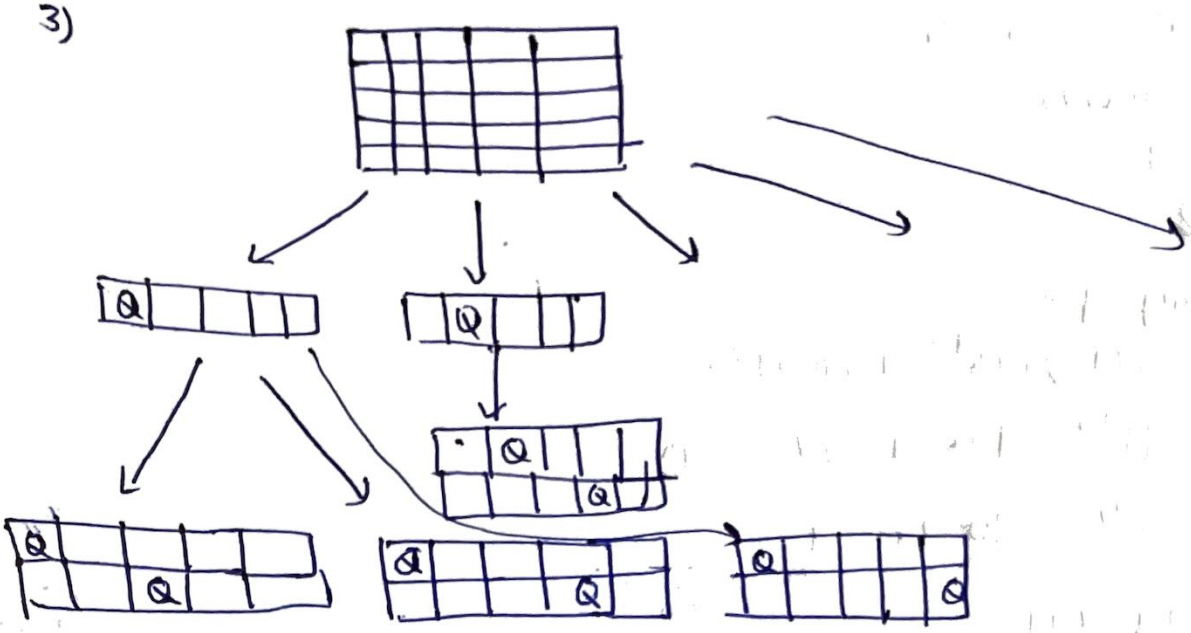
2) 1)



2)



3)



Yes it can be represented in the form of state space search

Yes it can also be solved by DFID.

But since its depth is fixed. DFID is not a good solution. Because all the depths need to be iterated to reach the final answer

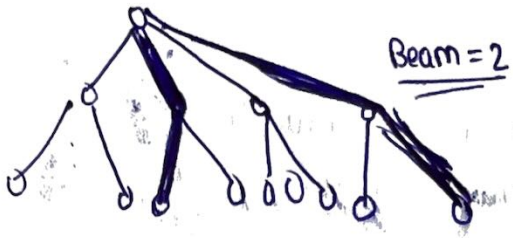
3)

Depth	Open	Close
	S	
1)	AB CD	S
2)	EJB CD	SA
3)	KJB CD	SAE
4)	FJB CD	SAEK
5)	JBCD	SAEKF
1)	BCD	SAEKFJ
2)	CD	SAEKFJB
1)	HD	SAEKFJB
2)	G	

Goal states →

4) Hill Climbing is a heuristic algorithm. It optimizes the solution.
It choose one best neighbour among all the neighbour.

Beam search chooses one or more best neighbours depending the width of the beam



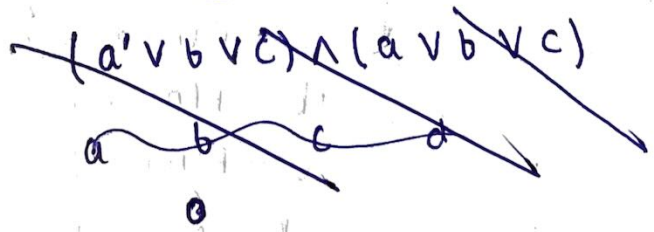
selects 2 best options

5) $(a' \vee d) \wedge (c \vee b) \wedge (c' \vee d) \wedge (d' \vee b') \wedge (a \vee d')$

a	b	c	d	c_1	c_2	c_3	c_4	c_5	h(n)
0	0	0	0	1	0	1	1	1	4
1	0	0	0	0	0	1	1	1	3
0	1	0	0	1	1	1	1	1	5 ✓
0	0	1	0	1	1	0	1	1	4
0	0	0	1	1	0	1	1	1	4

Ans (0 1 0 0)

6. $(a' \vee b \vee c) \wedge (a \vee b \vee c)$



7) Divide-and-conquer Frontier search

In DCFS we keep a tabu list of disallowed successors for each node that is added to Open.

The move generator is modified such that every time a node X (on open) is generated as a successor of some node Y, Y is excluded from being a successor of other node Z.

A relay layer is maintained to reconstruct the path. ~~Relay nodes contains the pointer to the address is a~~

Goal nodes contains the pointer to relay node. and DFS is used again to reconstruct the path

Sparse - Memory Graph search

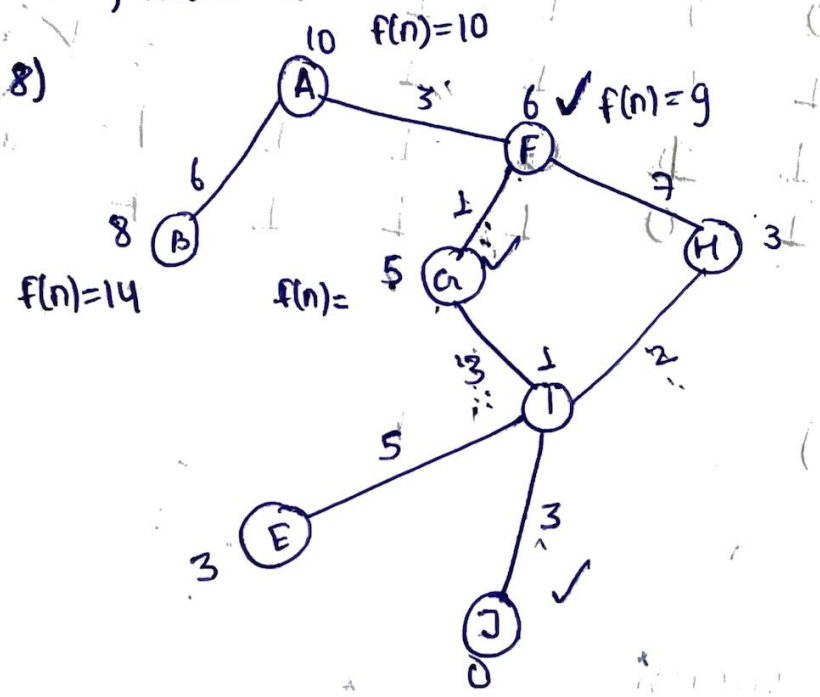
It identifies the boundary of the CLOSED list. The ~~boundary~~ boundary can be defined as those nodes on CLOSED that have one neighbour still on OPEN.

Kernels are the nodes with zero OPEN successor. Thus the algorithm prevents kernel nodes to be checked again. Thus preventing search from leaking back.

The SMAS also uses relay nodes for reconstruction of the path.

Once the PruneClosed is called all the boundary nodes are marked as relay nodes.

8)



✱ ✓ A → f(n) = 10

B → f(n) = 14

F → f(n) = 9 ✓

G → f(n) = 9 ✓

H − f(n) = 13

I − f(n) = 8

H − f(n) = 12

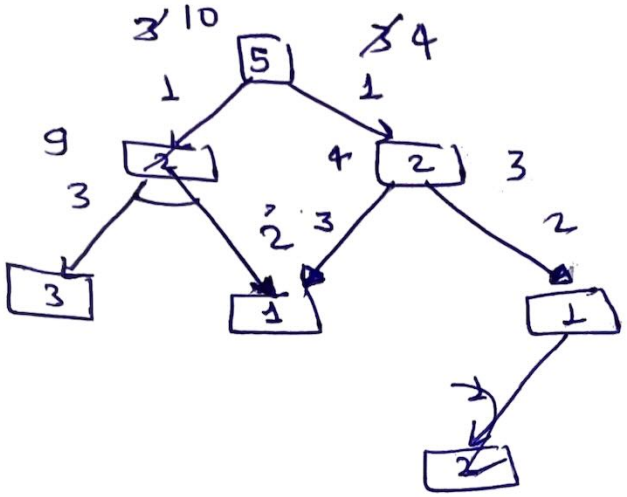
E − f(n) = 15

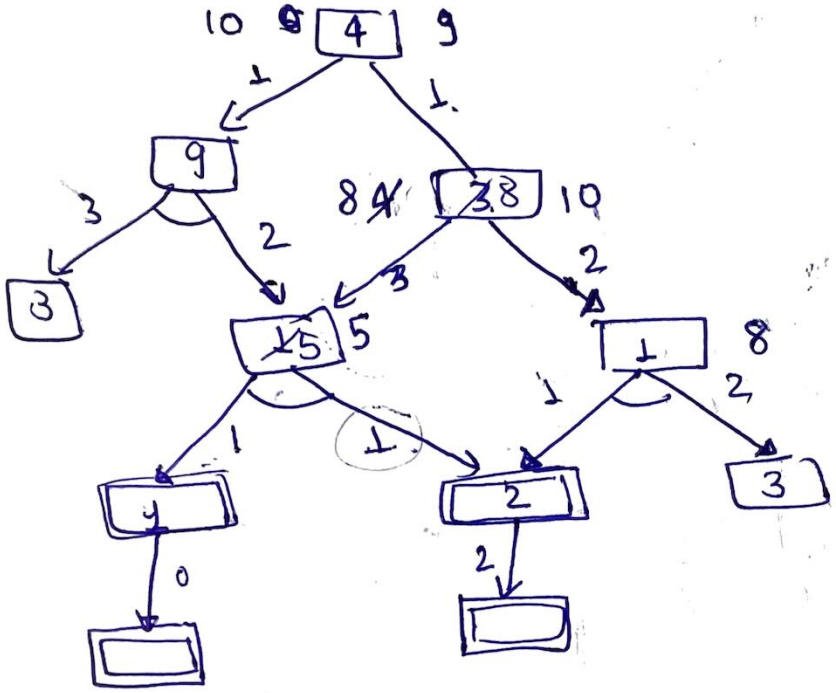
J − f(n) = 10

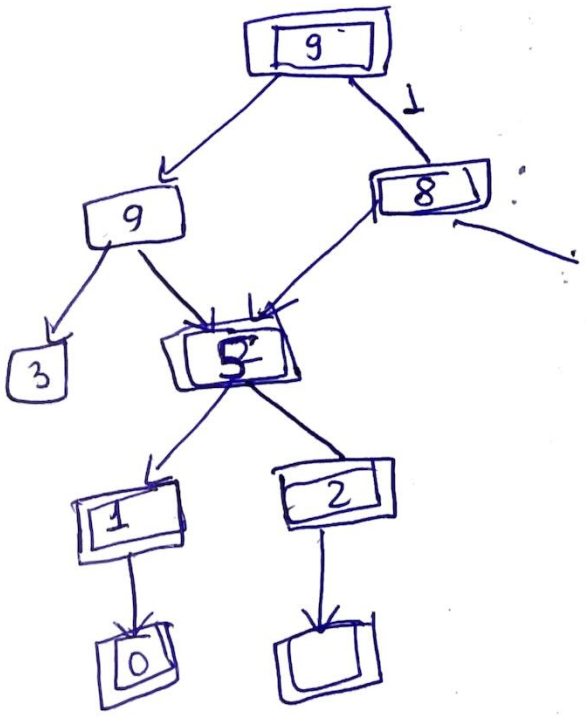
J = Goal

Path = A F G I J

9)


10
5
4
1
1
9
4
2
3
3
2
3
2
3
1
1
2


10
4
9
1
1
9
8
8
10
3
2
3
2
3
5
5
1
8
1
2
1
1
1
4
2
3
0
2


9
1
9
8
3
5
1
2
0

Ans ⇒ 9