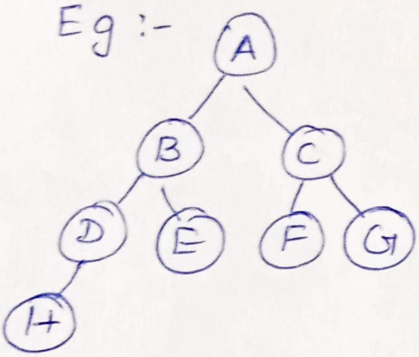
Iterative Deepening Depth first Search. (IDDFS)

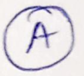
(*) IDDFS is the search algorithm which performs DFS iteratively.

(*) Here the DFS is run repeatedly with increasing depth limits until the goal is found.

Eg :-



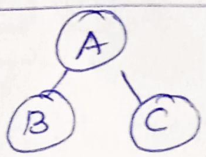
Iterative. 1



→ level 0

perform DFS

Iterative. 2

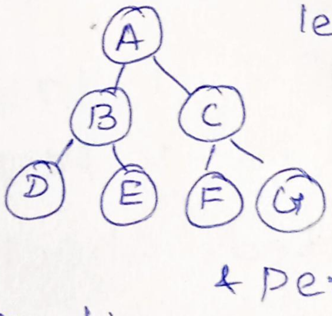


level 0

level 0

& perform DFS

Iterative. 3



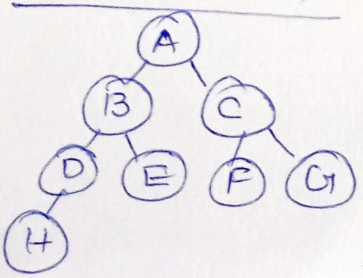
level 0

1

2

& perform DFS

Iterative. 4



level 0

1

2

3

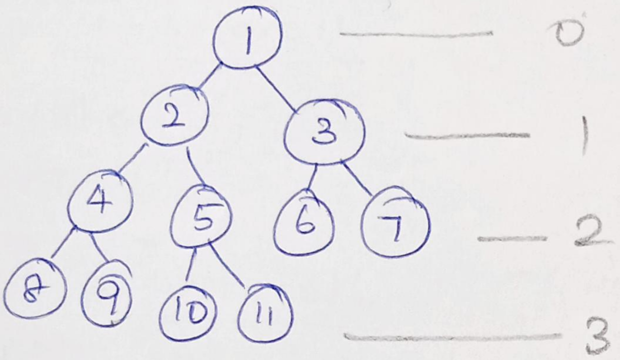
& perform DFS

* IDDFS combines DFS's space efficiently and BFS completeness (for finite path) pblms) to get solution path in less memory.

EG :-

Goal state : 9.

find its level



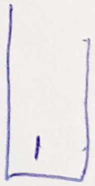
1

Sol :-

Iteration 1:-

→ 1 level 0 visited.

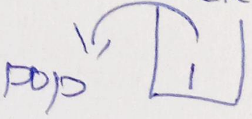
Stack status



olp seq = 1

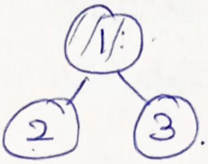
Perform DFS. mark 1 as visited. check adjacent ~~visited~~ node of 1. No node is visited so pop it

Stack status

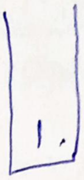


Stack is empty. So terminate. But we didn't reach our goal state. So perform 2nd iteration

iteration 2 :-

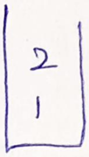


stack status



o/p seq :- 1

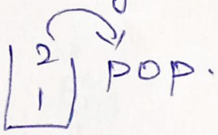
1 is visited. adjacent of node 1 is 2 & 3. push left child '2' first. visited 2.



1, 2

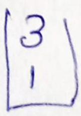
o/p seq :- 1 2

Adjacent of 2 is null. So pop. 2.



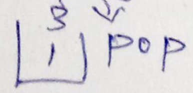
1

→ Now adjacent of 1 is '3' which is not visited. So mark it as visited and push 3 into stack



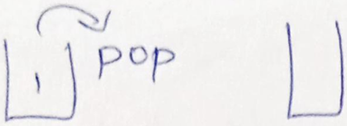
o/p seq :- 1, 2, 3

top of stack is 3 & Adjacent of 3 is null. So pop 3.



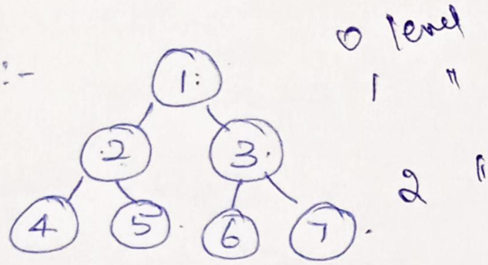
→ top of stack is 1, Now all node of 1 is visited so pop it.

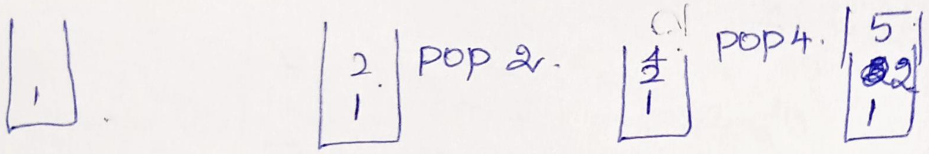
Stack



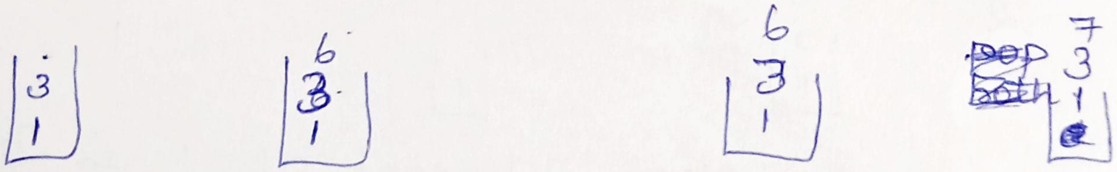
O|p seq : 1,2,3. stack is empty and in Iteration 2 also we didn't reach goal ~~sa~~ state. So perform iteration 3.

iteration 3 :-

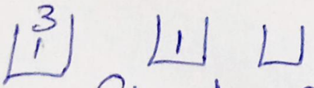




O|p:-seq :- 1   1, 2   1 2 4   1 2 4 5

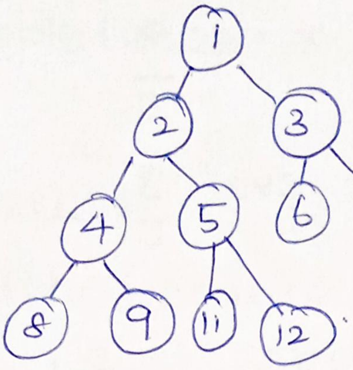


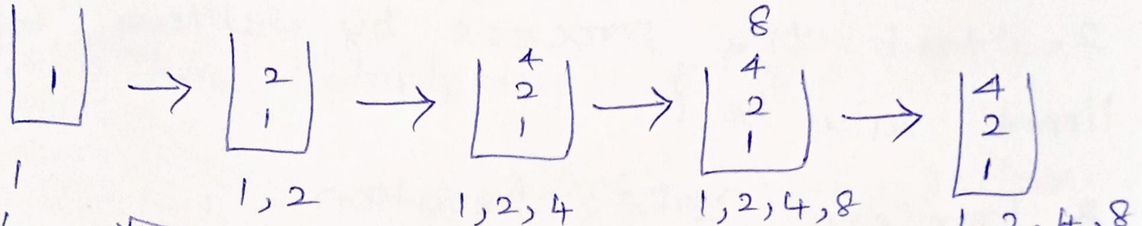
1 2 4 5 3   1 2 ~~3~~ 4 5 3 6   1 2 4 5 3 6 7



Goal state didn't reach. So perform iteration 4.

Iteration 4





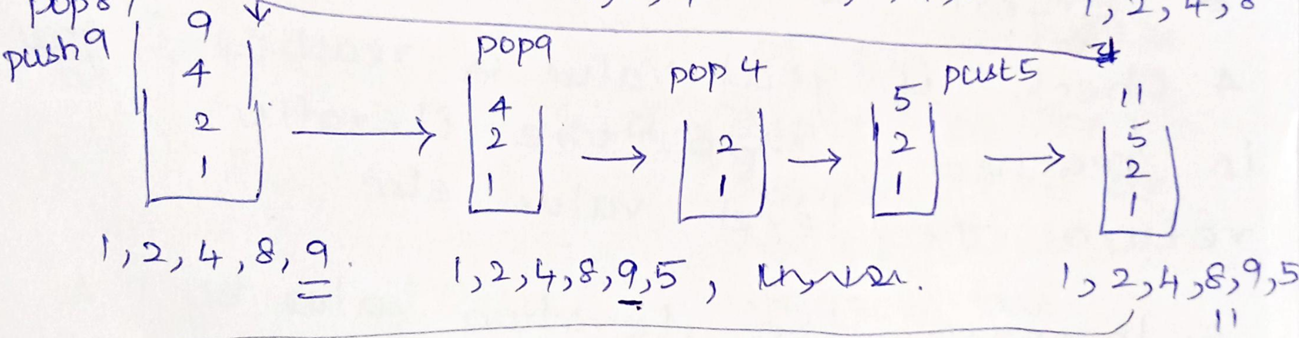
1

1, 2

1, 2, 4

1, 2, 4, 8

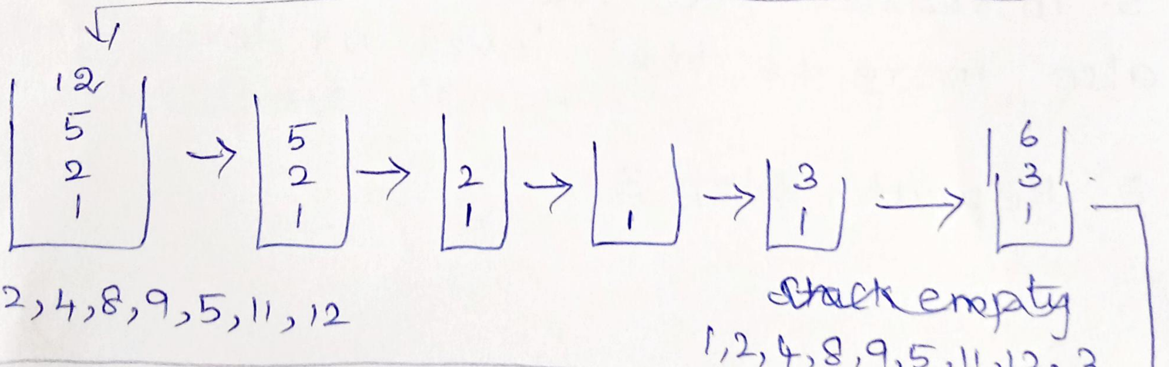
1, 2, 4, 8



1, 2, 4, 8, 9

1, 2, 4, 8, 9, 5, [illegible].

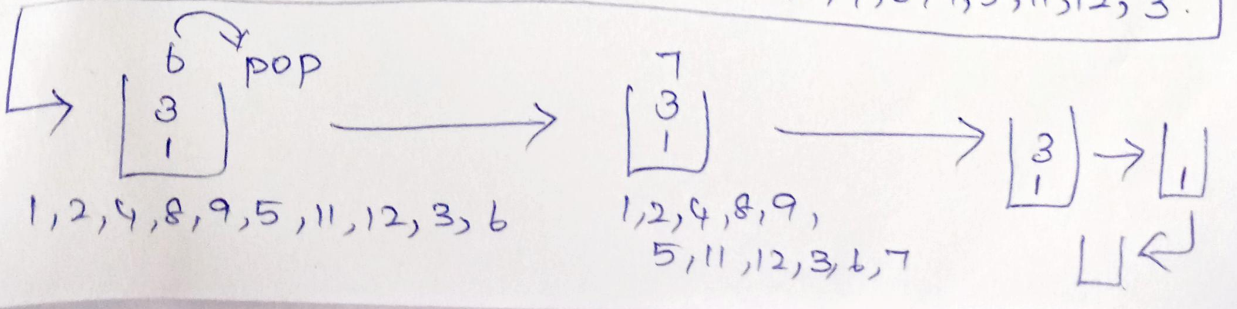
1, 2, 4, 8, 9, 5, 11



1, 2, 4, 8, 9, 5, 11, 12

stack empty

1, 2, 4, 8, 9, 5, 11, 12, 3.



1, 2, 4, 8, 9, 5, 11, 12, 3, 6

1, 2, 4, 8, 9, 5, 11, 12, 3, 6, 7

O/p seq:

1 2 8 4 $\underline{\underline{9}}$ 5 11 12 3 6 7

Goal state is in : level. 3.

AlG :-

1. Initiate the iteration value as 1
2. Start the process by setting level limit value to 0.
3. Perform DFS operation
4. Check if goal value is reached, if goal is reached stop the iteration & return the level value else
5. Increase the iteration value by 1 & also increase the level or level limit by 1
6. Repeat step 3.

Performance Evaluation :-

(1) completeness → IDDFS is complete as it gives solution

(2) optimalitity :- It does not have optimal sol. because it stops when the first goal node is reached.

(3) Time Complexity :- $O(b^d)$

b → branching factor. or. no. of nodes

d → depth of the search tree or. no. of levels in search tree

(4) Space complexity : $O(bd)$