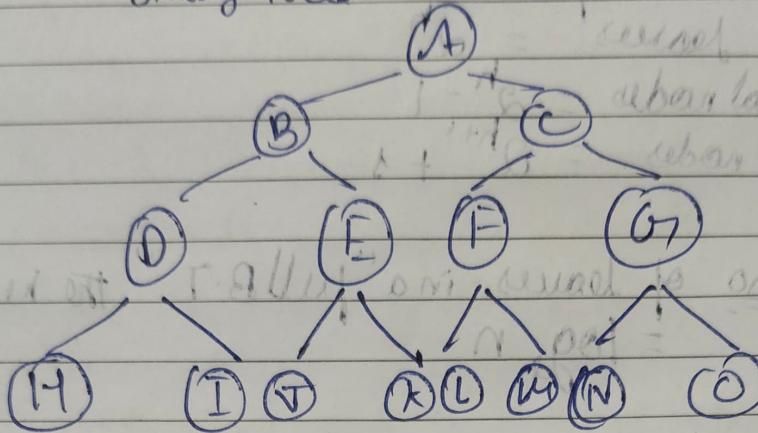


Time and Space Complexity

Date _____
Page _____

~~BFS~~

Binary tree



$$\begin{array}{l|l} b = \text{branching Factor} = 2 & \text{Start} = 1 \\ m = \text{depth (max)} = 3 & \text{Goal} = 0 \end{array}$$

BFS:

Space :- For BFS we use Queue. When we explore a node we get their children added to the queue. So at one point all nodes in the lowest level get added & we need certain amount of space. It depends on 'b'. So the max no. of nodes stored is $O(b^m)$

added children	A B C D E F G H I J K L M N O
explore nodes	A B C D E F G

Time :- It is the sum of all the space complexities as we have to explore each node for it ie $b^0 + b^1 + b^2 + \dots + b^m$ This is a geometric progression

$$\left(\frac{b^{m+1} - 1}{b - 1} \right)$$

But in the last level leaves dominate so we take it as $O(b^m)$

This is the max no. of nodes to travel if we have a full tree.

~~Time~~: For DFS we use stack. we explore a node then its left child & same until we reach a leaf & back track to previous node & track its right child & continue the same.

So the time complexity remains the same as BFS
 $O(b^m)$

For space when we reach a leaf we back track & it repeats to each leaf so for each cycle m no of nodes are stored & it is the same for all cycles

Cycle 1 :- AB O H

Cycle 2 :- AB O C

Cycle 3 :- AB E J & so on

so max space required is $O(bm)$