

-> "You don't need anything if God is on your side." - RAMRAM

-> Prerequisite:->

<https://drive.google.com/file/d/12khoQ6f2H1YLIrKdE2eyehpJwb1BHkF/view?usp=sharing>

Doc ->

https://docs.google.com/document/d/178xP-4NLVg-160QyPYxbdVzLjcCow3X46fdd83HT_H0/edit

-> Today I am going to reveal the tricks I learnt over a period of 2 years!

How does the DFS on tree looks like ?->

Trick 1 Just do the boundary traversal of the tree from left to right and you will get the DFS order.

Observation1 You start from node 1 and you also end at node1.

-> How to solve DP on Tree problems. ?

No matter what the question is; you should always calculate the answer of bottom nodes ; then only you should go up

Trick 1 Always calculate the answer for bottom nodes ; then start going up ; this really helps ; because the answer of bottom nodes is used to calculate the answer of nodes which are coming up (above!)

Tanvir Singh Law :-> answer to a node can only be calculated when the answer has been calculated for the children.

Q : Given a Tree of "N" nodes; find the height of each node and print it ; the tree is rooted at Node-1.

Height of node x means the longest path you can go from a current node x downwards.

Tiju Law -> Longest distance from current node to some leaf node

$$\text{Height}[i] = 1 + \max(\text{height}[c1], \text{height}[c2], \dots, \text{height}[ck])$$

Where; $c1, c2, \dots, ck$ are all the children of node "i"

-> We don't know how to travel the tree from bottom to top

-> Anwesha's Law

: bottom up traversal: print node in dfs code after the for loop

-> <https://ideone.com/ZZgRHc>

C++ <https://ideone.com/FHLOF4>

Java <https://ideone.com/wfkQzy>

Py <https://ideone.com/5thjxI>

<https://ideone.com/5thjxI>

-> In DFS ; you come outside the first for loop only when you reach a leaf node for the first time; hence printing the node after the first loop guarantees that all the leaves will be printed first

P1 : - Given a Tree of N Nodes, rooted at node 1 find the sum of each subtree "i" in the given tree.

$\text{Sum}[i] = \text{value of node "i"} + \text{sum}(\text{sum}[c1] + \text{sum}[c2] + \text{sum}[c3] + \dots \text{sum}[cg]) = \text{sum of subtree rooted at node 'i'}$

—> c1,c2,.....cg:-> children of node "i"

Algorithm.-> <https://ideone.com/qcAhBa>

(Google Interview Problem)

P1 :- Given a Tree of N Nodes, rooted at node 1 find the maximum sum of subtree possible.

Sol : - Find the sum of all the subtrees and find the maximum.

TC : $O(N)$ ---->

SC : $O(N)$ → Sum array

→ $O(N)$ → for the graph of tree → you can ignore this depending on the interviewer

→ Recursion takes $O(N)$ space in the worst case; so you may consider that too → $O(\text{height})$; but in the worst case; height = N .

C++ <https://ideone.com/W8t1yD>

Java. <https://ideone.com/WJpxHR>

Python. <https://ideone.com/saVwGT>

```
1. void DFS(int node,vector <int> G[],int used[],int parent[]){
2.
3.     used[node] = 1 ;
4.
5.     for(auto u: G[node]){ //iterating all children "u" of "node"
6.
7.         if(used[u]==0){
8.             //if this node/branch has never been visited before
9.             //just go into it and search it using dfs in recursion
10.            parent[u] = node ;
11.            DFS(u,G,used,parent);
12.
13.        }
14.    }

    print(node);
15. }
```