DP on Trees

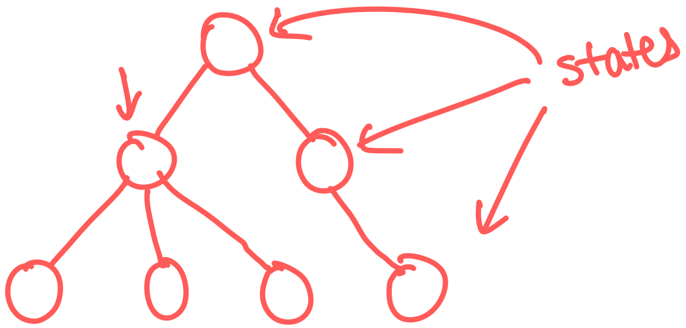
- Raghav Goel

Goal

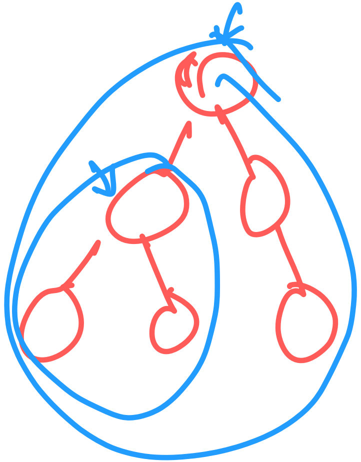
- DP on Trees
- Rerooting

DP on Trees → Subtree DP

- DP states are defined for subtrees rooted at a node.
- **State:** dp[node] stores information for the subtree rooted at node.
- **Transitions:** Aggregate results from child nodes using recursive relations.
- **Example:** Finding the size of each subtree.



Tree



DP

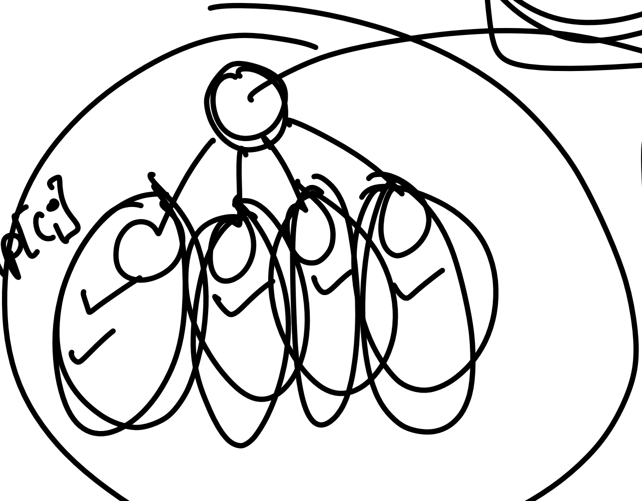
~~States~~
transitions

each node
⇒ size of the subtree

~~Subtree DP~~

$dp[node] \Rightarrow$ number of nodes in the subtree

dp[ci]



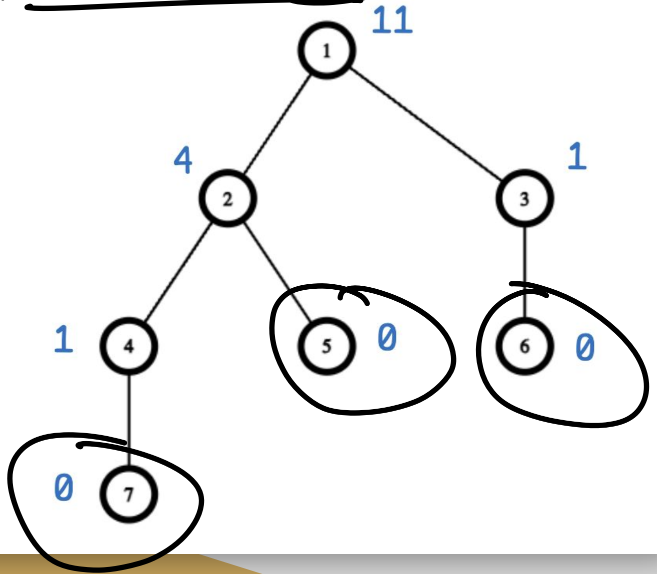
$$dp[node] = 1 + \sum_{c \in children} dp[c]$$

Problem: Sum of Distances in Subtree

Given a tree with ***N*** ($1 <= N <= 10^5$) nodes where the nodes are numbered from 1 to *N*. The tree is rooted at node 1.

For each node, print the **sum of distances** to **all nodes in its subtree**.

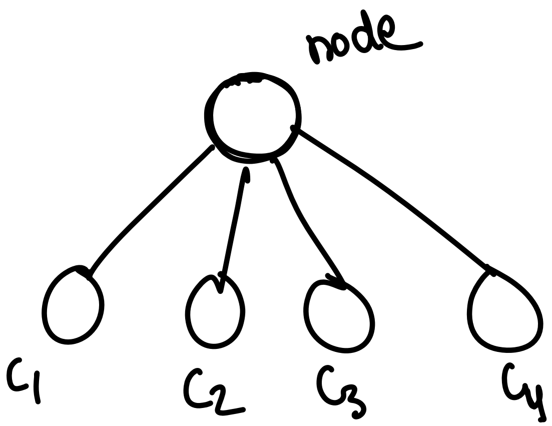
subtree DP



Example:

The answer for each node is written in blue.

DP

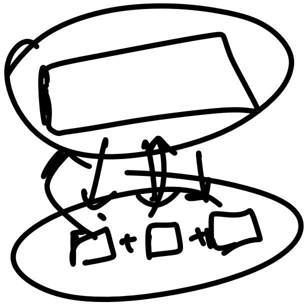


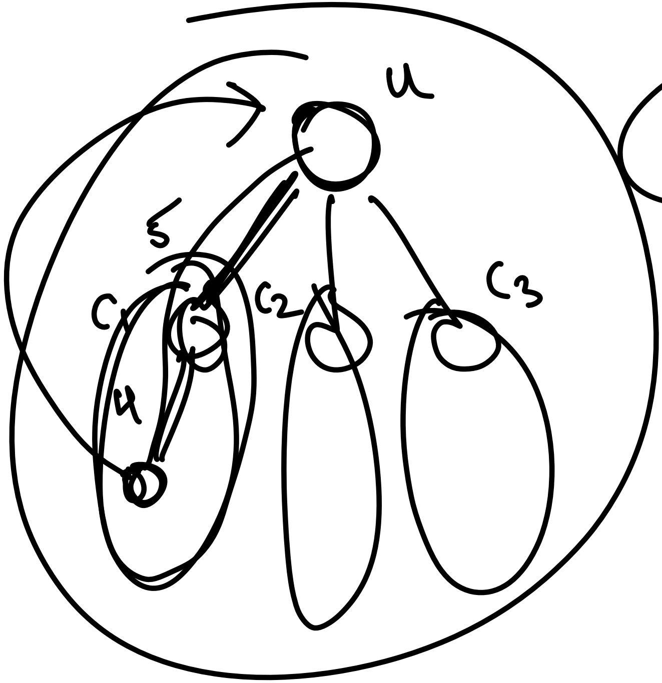
State

$dp[node]$ = sum of distances from node to all nodes in the subtree

$dp[node]$

$(dp[c_1])$ $(dp[c_2])$ $(dp[c_3])$ $(dp[c_4])$





$$dp[u] = \sum_{v \in subtree(u)} dist(u, v)$$

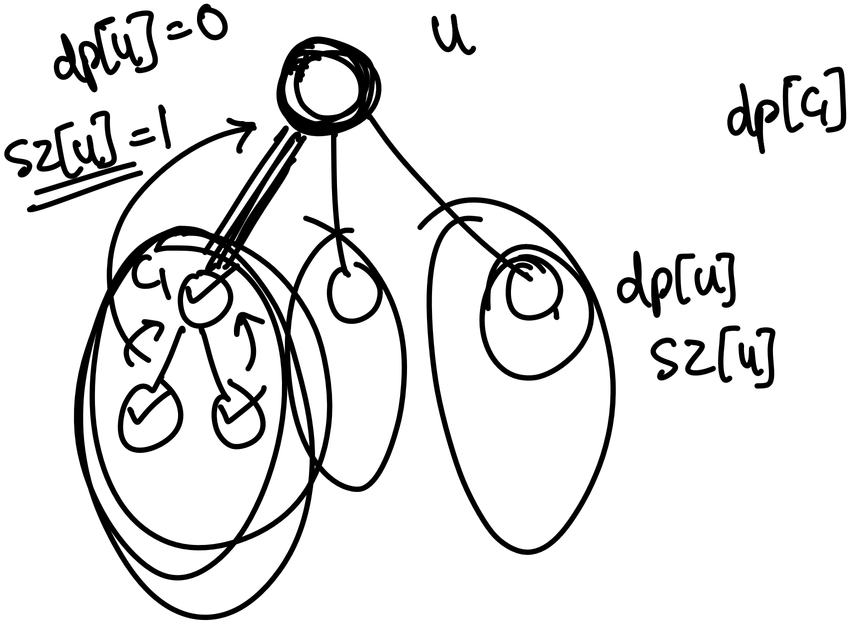
$$dp[c] = \sum_{v \in subtree(c)} dist(c, v)$$

$$dist(u, v) = dist(c, v) + 1$$

$v \in subtree(c)$

dp[c]

dp[u]



$$\boxed{\boxed{dp[u]} = \sum dp[c_i] + \boxed{sz}[c_i]}$$

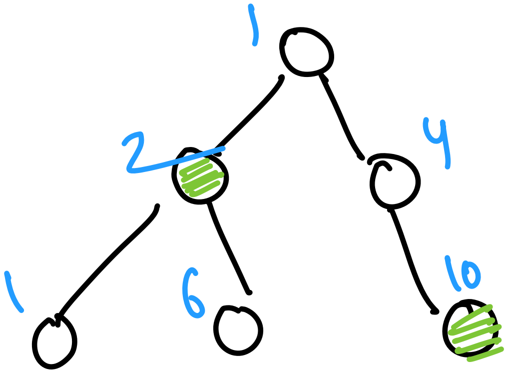
dp

sz

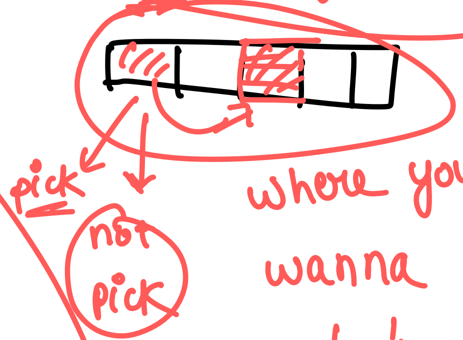
Problem: House Robber 3

Given a tree of $\boldsymbol{N}$ ($1 <= N <= 10^5$) nodes, where the $\boldsymbol{i}$-th node has $\boldsymbol{C_i}$ ($1 <= C_i <= 10^4$) coins. You need to choose a subset of nodes such that every pair of nodes in this subset is not connected by an edge. Find the maximum number of coins you can get.

Problem Link - https://leetcode.com/problems/house-robber-iii



subset of indices



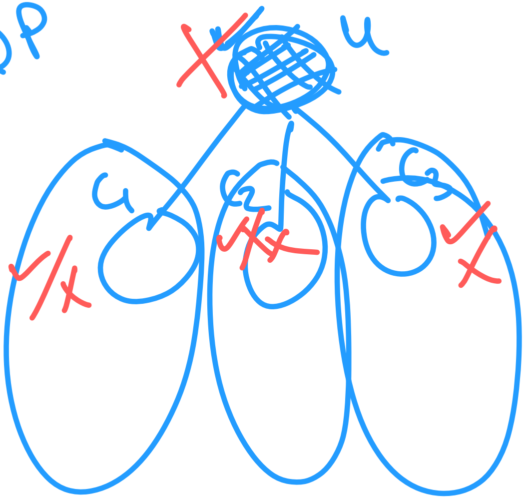
where you wanna maximize $\sum a_i$

$dp[i][0/1] \rightarrow [1 \dots i]$

$0 \rightarrow$ not pick

$1 \rightarrow$ picked

DP



$dp[u][0] \rightarrow$ max. ans that we can get if we don't pick u.

$dp[u][1]$

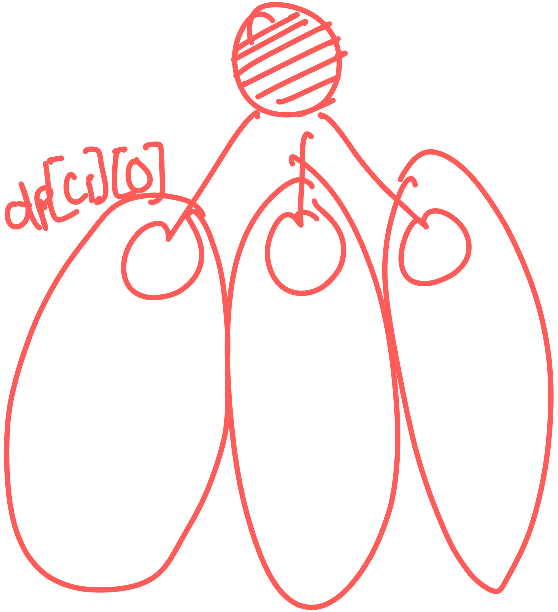
$\downarrow$

max. ans (pick)

$$\boxed{dp[u][1] = c[u] + \sum_{c_i \in \text{children}} dp[c_i][0]}$$

$$\boxed{dp[u][0] = \sum_{c_i \in \text{children}} \max(dp[c_i][0], dp[c_i][1])}$$

pick

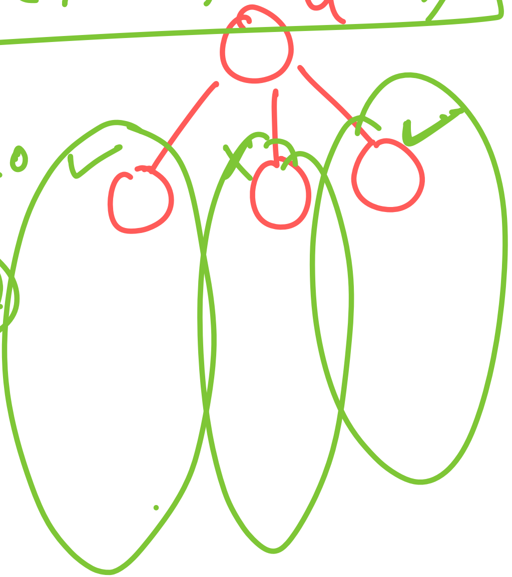


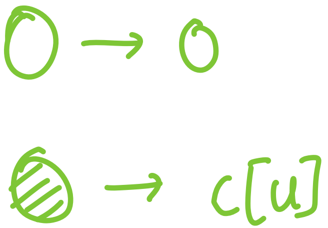
not pick

$$dp[u][0] = \sum \left(\max(dp[c_i][0],\ dp[c_i][1])\right)$$

$dp[c_i][1] \to 10$

$dp[c_i][0] \to 12$



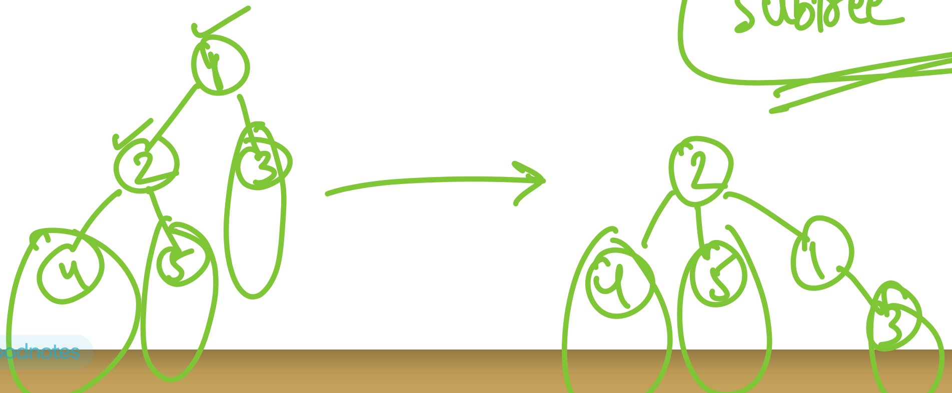


Rerooting

Rerooting is a technique where we calculate results for every node in the tree by systematically "re-rooting" the tree at each node and reusing previously computed subtree results to maintain efficiency.

- find something for each node

subtree DP for root



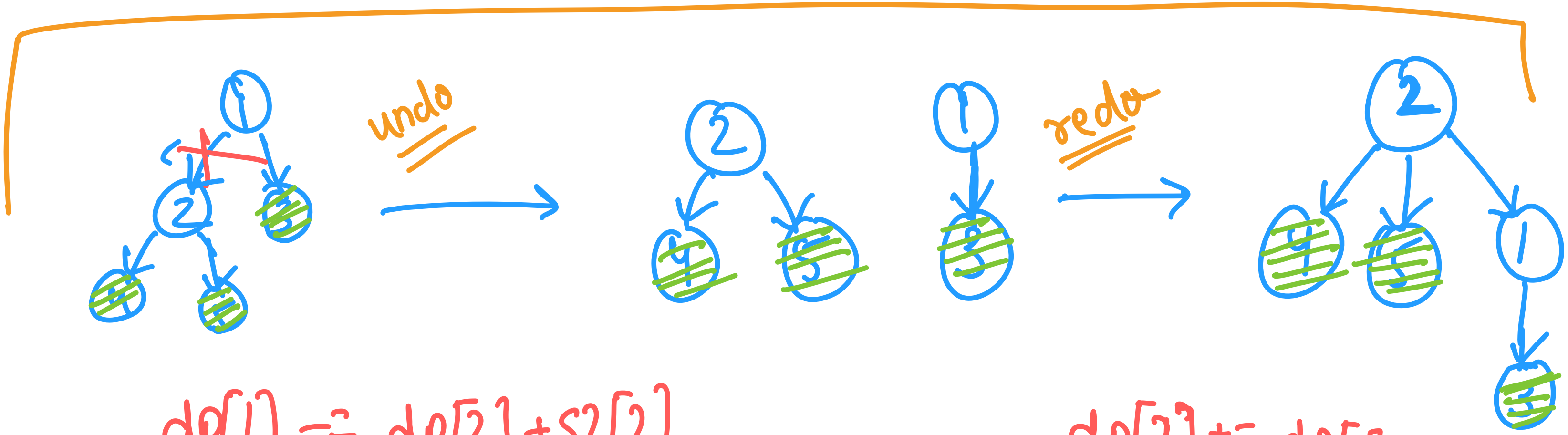
subtree dp

$dp[u] \rightarrow$ sum of distances to all node in the ~~subtree~~ tree

$dp[root] \rightarrow$ ans

changeRoot(oldRoot, newRoot)

dfs



$dp[1] -= dp[2] + sz[2]$

$sz[1] -= sz[2]$

$dp[2] += dp[1] + sz[1]$

$sz[2] += sz[1]$

→ Problem → find some information for all nodes

↓

info for one node

↓ subtree dp

calculated dp states for all nodes

where dp[node] → some info for subtree of node

←

check if changeRoot (undo, redo)

←

Rerooting DP

Steps for Rerooting

- **Compute Results for One Root:**
  - Solve the problem for the root using **Subtree DP**.
- **Shift the Root:**
  - Using the result from the first root, **shift the root** to each of its children without recomputing everything from scratch.
  - Update the result for the new root based on the relationship between the old root and the new one

Problem: Tree XOR

D. Tree XOR

time limit per test: 3 seconds
memory limit per test: 512 megabytes

You are given a tree with n vertices labeled from 1 to n. An integer a_i is written on vertex i for $i = 1, 2, \ldots, n$. You want to make all a_i equal by performing some (possibly, zero) spells. 1

Suppose you root the tree at some vertex. On each spell, you can select any vertex v and any non-negative integer c. Then for all vertices i in the subtree[†] of v, replace a_i with $a_i \oplus c$. The cost of this spell is $s \cdot c$, where s is the number of vertices in the subtree. Here $\oplus$ denotes the bitwise XOR operation.

Let m_r be the minimum possible total cost required to make all a_i equal, if vertex r is chosen as the root of the tree. Find $m_1, m_2, \ldots, m_n$.

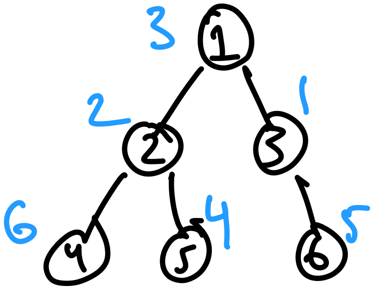
[†] Suppose vertex r is chosen as the root of the tree. Then vertex i belongs to the subtree of v if the simple path from i to r contains v.

Link: https://codeforces.com/contest/1882/problem/D

- some info for all nodes

↓

one node

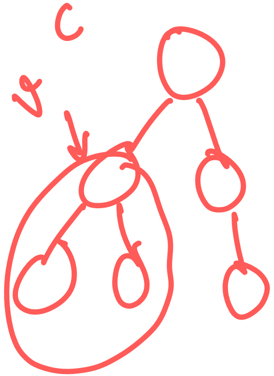


Subtree DP

Tree

(1)

all a_i equal



$\boxed{a_i} = c$

$i \in \text{subtree}(v)$

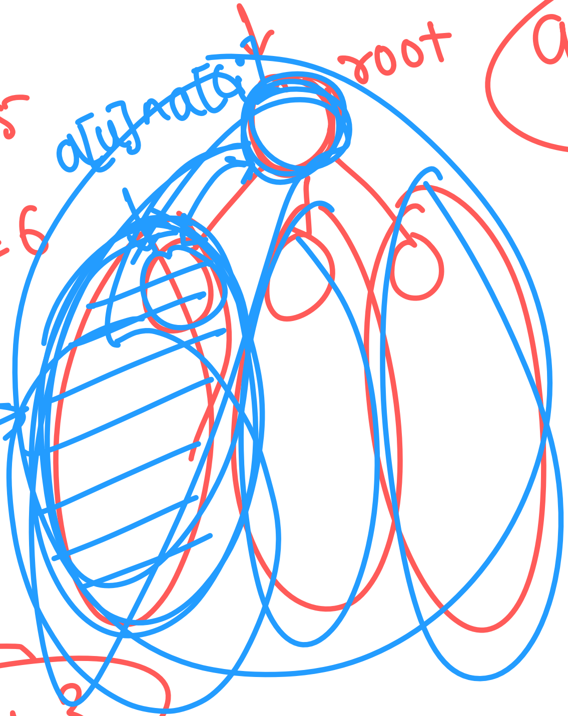
$sz[v] * c$

$a[u] = 5$

$a[c_1] = 6$

$a[u] \wedge a[c_1]$

root



$sz[c_1] * 3$

$\downarrow$

$(2+1)$

$a[c_1] \to 5$

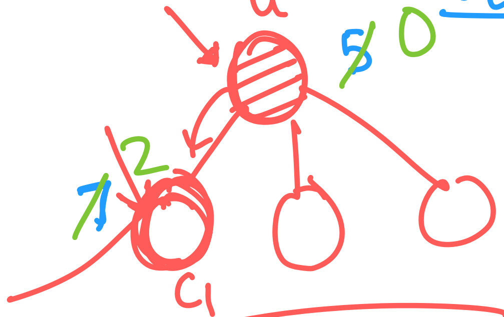
$c = 5$

$\wedge = 3$

$\longrightarrow$

$a[u] \Rightarrow a[p]$

~~$a[u] \wedge = a[u] \wedge a[p]$~~

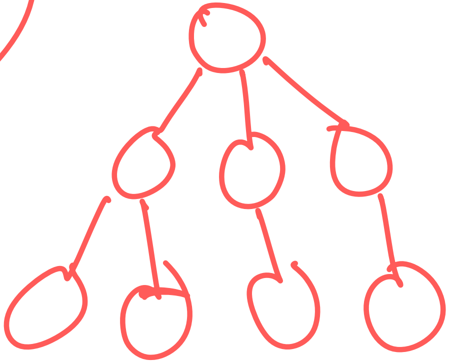


$c = a[u] \wedge a[c_1]$

$a[u] = 5$

$a[c_1] = 6$

Subtree DP



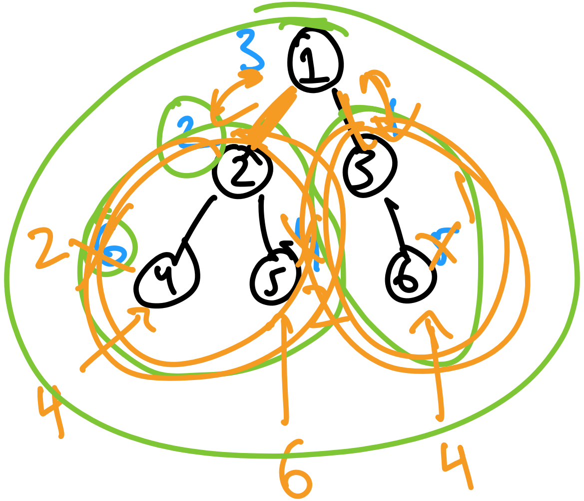
$$\boxed{ans[root] = \sum_{u} \sum_{c_i} (a[u] \wedge a[c_i]) * sz[c_i]}$$

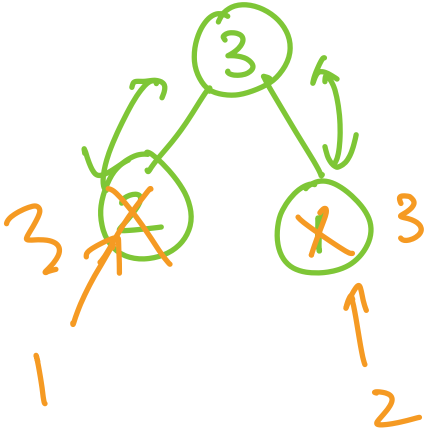
$f(u, c_i)$

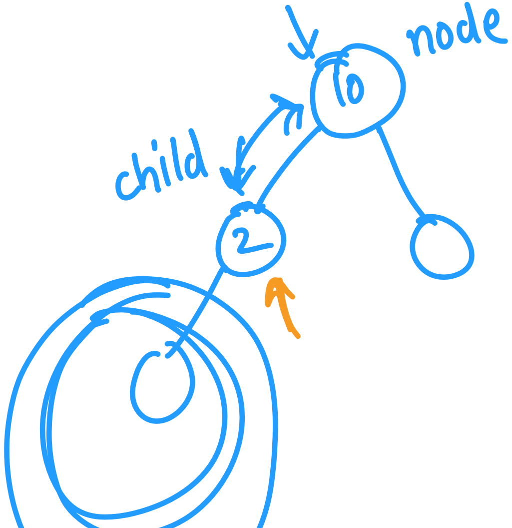
$$\boxed{dp[u] = \sum \left(f(u, c_i) + dp[c_i] \right)}$$

$$ans[root] = \sum (a[u] \wedge a[p]) * sz[u]$$

$2 \wedge 6 \Rightarrow 4$







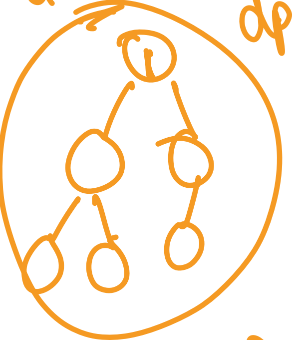
u

$c = a[u] \wedge a[p]$

$$\sum_u \underbrace{(a[u] \wedge a[p]) * sz[u]}$$

$$\boxed{ans[root] = \sum_u cost(p,u)}$$

$ans[i] = dp[i]$

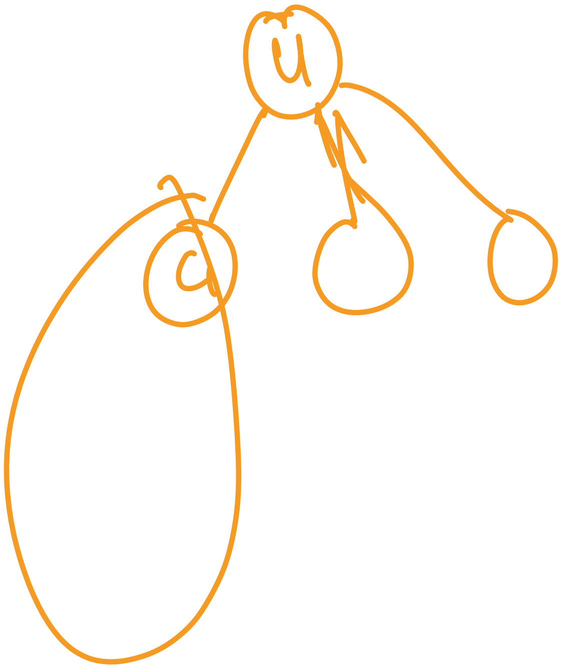


$cost(p,u)$

subtree DP

$$\boxed{dp[u] = \sum_{c_i} \left((a[u] \wedge a[c_i]) * sz[c_i] + dp[c_i] \right)}$$

1 subtree DP



$$dp[u] \cancel{+=} -= (a[u] \wedge a[c_1]) * sz[c_1] + dp[c_1]$$

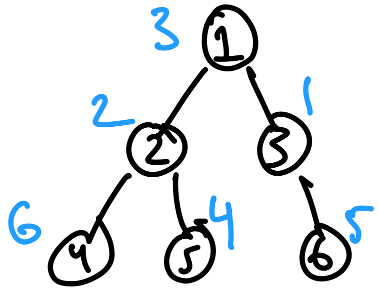
undo easy

$$sz[u] \cancel{+=} -= sz[c_1]$$

Rerooting

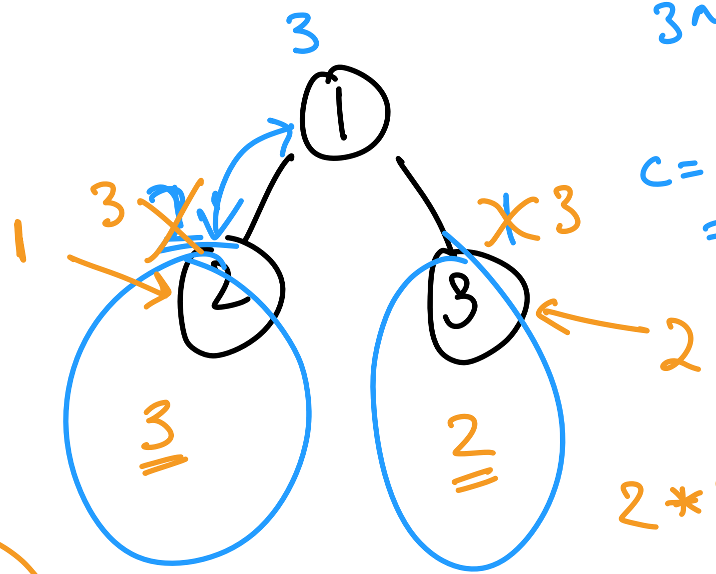
observation transition

problem → all nodes → 1 node → subtree DP → undo not → Rerooting



3^2

$c = 3^1$
$= 2$



$1 * 3 = 3$

$2 * 2 = 4$

Kartik Arora

DP on Trees

Homework

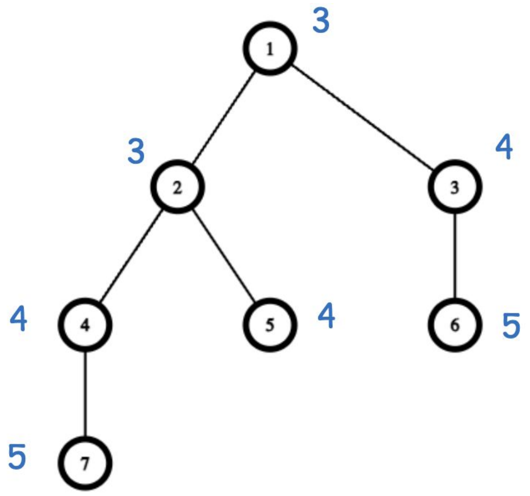
Problem: Maximum Distance to any node

Given a tree with ***N*** $(1 <= N <= 10^5)$ nodes where the nodes are numbered from 1 to *N*.

For each node, print the **maximum distances** to **any node in the tree**.

Example:

The answer for each node is written in blue.



Homework:-

- Subtree DP problems

  redo all problems discussed in class

- Rerooting:-
  - redo first problem (spend good enough)
  - attempt the second problem