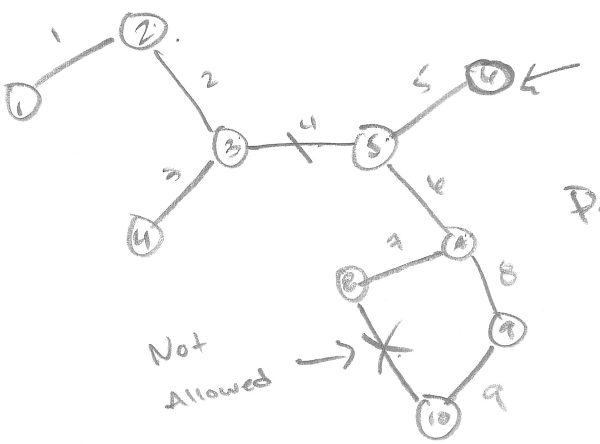
7-16 : Trees

- Tree has both of the following:
  1) If a graph has n nodes, it has n-1 edges
  2) It is acyclical



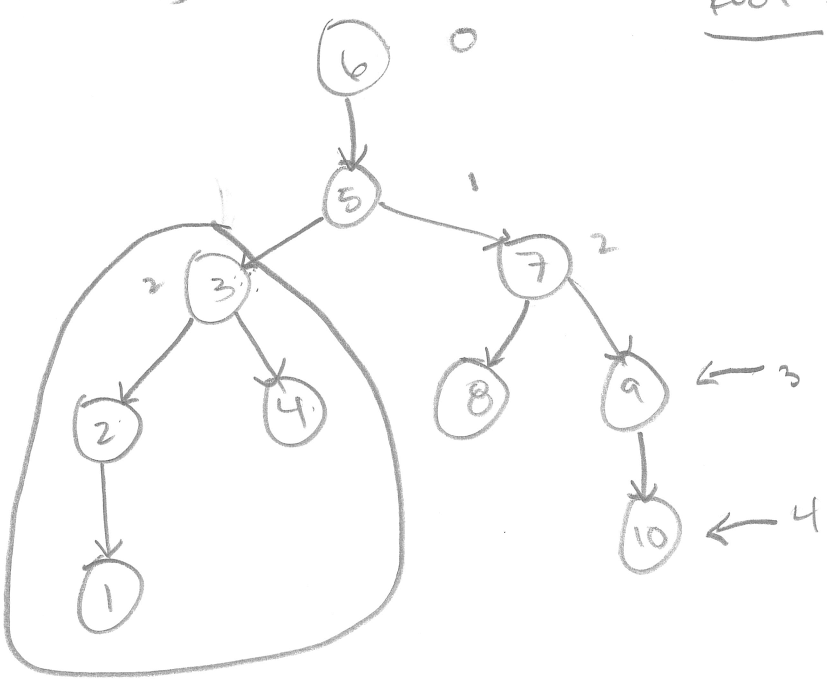
→ Exists exactly 1 simple path btw. any 2 nodes.

Path from node 2 to node 7

2 → 3 → 5 → 7

- Rooting at node 6:

Root: The single node w/ no incoming edges.

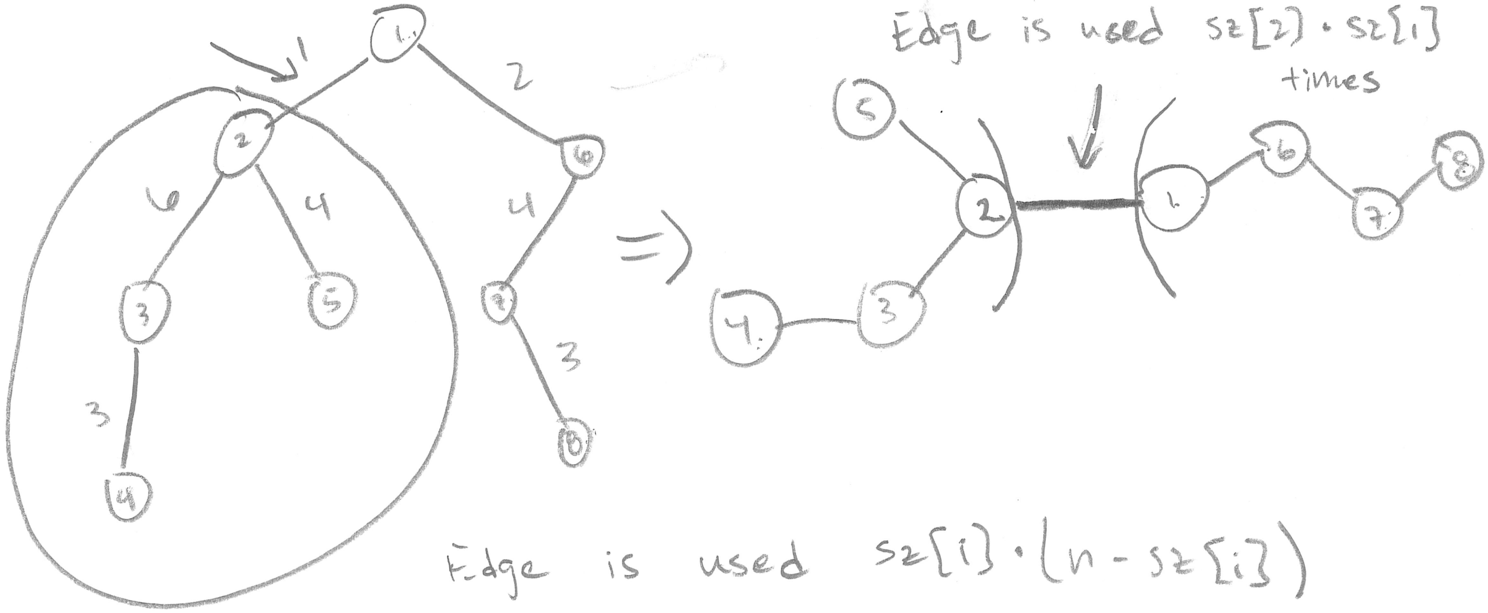


sz[3] = sz[2] + sz[4] + 1

itself

Caves:

?



res = 0
For each i
→ res += w_i * (sz[i] · (n - sz[i]));

return res/2;

Tree Diameter:

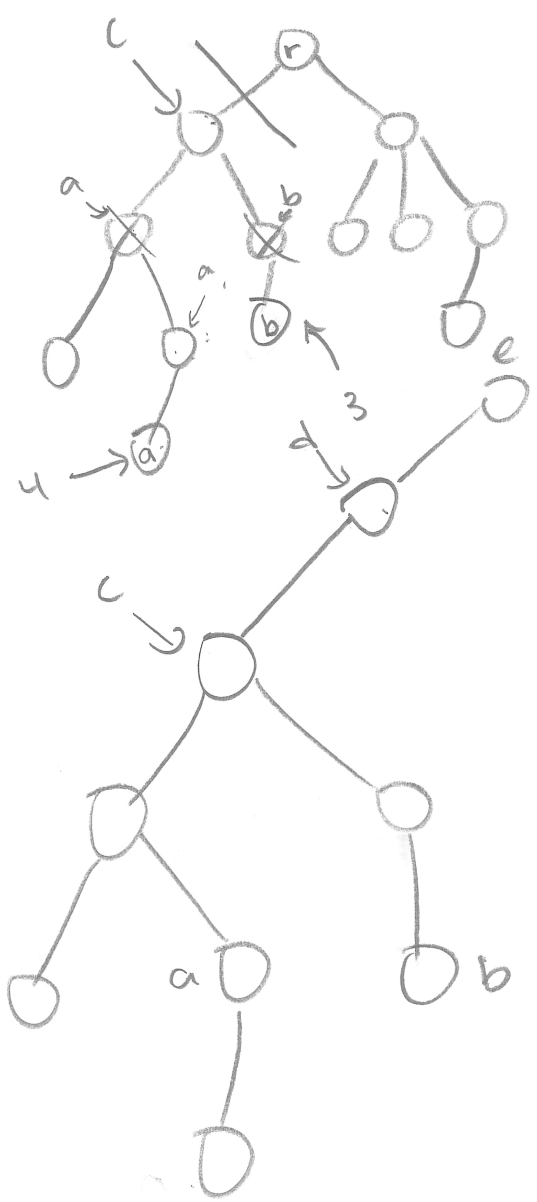
Def: Longest path btw. any 2 nodes in the tree.

To get tree diameter:

1) Run BFS/DFS from arbitrary node S_0
   s_1 → Get furthest node from s_0 from the DFS
2) Run DFS from s_1 to find furthest node (s_2)

3) Diameter is path from s_1 → s_2

LCA (Lowest Common Ancestor) - O(n):



1) DFS from root to calculate depths of all nodes

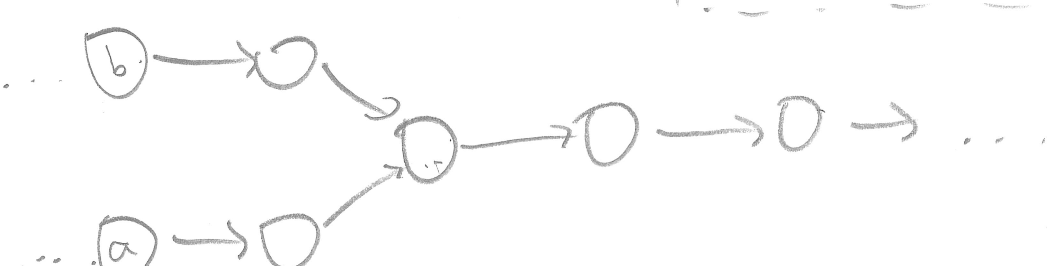
2) Ascend up the tree from deepest node until a & b at same depth

3)
```
while (a != b) {
    a = parent[a];
    b = parent[b];
}
```

4) a (or b) = LCA

2

```
while (d[a] > d[b]) a = p[a]
while (d[b] > d[a]) b = p[b]
```



→ Binary Lifting

→ Centroid Decomposition

→ Tree DP

→ Euler Tour / Tree Flattening

```
for i in range (1,10000)
    for j in range(1, 10000/i)
```

$$\Sigma = \frac{10000}{1} + \frac{10000}{2} + \frac{10000}{3} \cdots \frac{10000}{10000}$$

$$= 10000\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{10000}\right)$$

$$= 10000\left(\underbrace{1 + \underbrace{\frac{1}{2} + \frac{1}{2}}_{1} + \underbrace{\frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}}_{1} + \underbrace{\frac{1}{8} + \cdots +}_{1}}_{15}\right)$$

$$= 10000 \lg(10000)$$