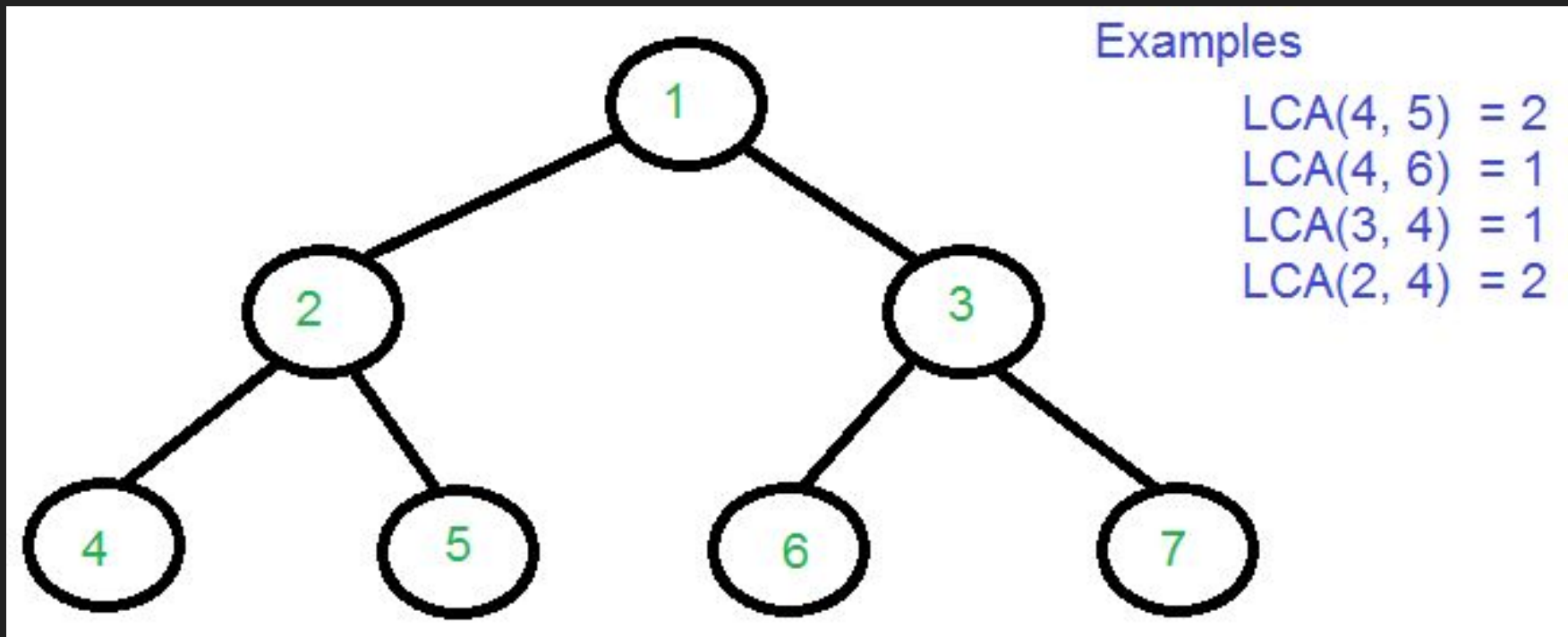


What is LCA



Subproblem:

Find the k th parent of a node in a tree.

Binary Lifting !!!

Let's express k in binary $k = 11$

$$11 = 2^3 + 2^1 + 2^0$$

Let's say we are at node u.

$V1 = 8^{\text{th}}$ parent of u

$V2 = 2^{\text{nd}}$ parent of v1

$V3 = 1^{\text{st}}$ parent of v2

V3 is our answer.

So, let's say we store the $(2^k)^{\text{th}}$ parent for each node in the tree. How much memory are we using?

$N \log N$

If we have N nodes, then we only have to store k parents such that $2^k \leq N$. So, $k \leq \log N$ for each node. Hence, $N \log N$

Now, after this preprocessing we can get the $(2^k)^{\text{th}}$ parent of each node in $O(1)$.

So, in order to find k th parent of any node, we need to do at most $\log K$ operations.

Pre-processing for binary lifting

```
//Assume you have the just immediate parent already.  
//If you don't have that, you can run a dfs and get that.  
for (int i = 1; i < limit; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        //2^ith parent of node j  
        int x = parent[j][i - 1]; // x = 2^(i-1)th parent  
        parent[j][i] = parent[x][i - 1]; // 2^ith parent = 2^(i-1)th parent of x  
    }  
}
```

Finding k^{th} parent of a node

```
|  
int kth_parent(int node, int **parent, int k) {  
    if (k == 0)  
        return node;  
    int log_val = log2(k);  
    int max_power = pow(2, log_val);  
    return kth_parent(parent[node][log_val], parent, k - max_power);  
}
```

Practice Problem

<https://cses.fi/problemset/task/1750>

Finding the LCA

```
int find_lca(int a, int b, int *level, int limit)
{
    if (level[a] > level[b])
        swap(a, b);
    int d = level[b] - level[a];
    b = kth_parent(b, parent, d);

    //now a and b are at same level
    if (a == b) // return a if both are at same node
        return a;

    for (int i = limit - 1; i >= 0; i--)
    {
        //try to move the nodes a and b upwards till they both are not equal
        if (parent[a][i] != -1 && (parent[a][i] != parent[b][i]))
        {
            a = parent[a][i];
            b = parent[b][i];
        }
    }
    //now a and b's just next parent is their LCA
    return parent[a][0];
}
```


Distance Queries

Given 2 nodes A and B, find the distance between them.

- 1) Unweighted Tree
- 2) Weighted Tree

Problems

Counting Paths:

[https://cses.fi/problemset/task/1136\](https://cses.fi/problemset/task/1136)

Video Editorial:

<https://www.youtube.com/watch?v=8Nq3THy2Kw0>