Trees 2

Diameter

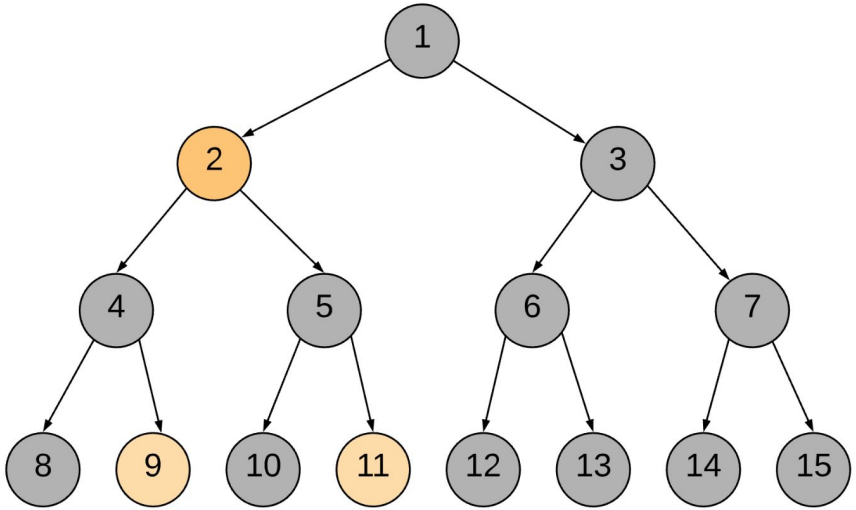
- Longest path present in the tree is the diameter
- Useful technique to think about problems involving trees

LCA

- Lowest Common Ancestor
- For a rooted tree LCA of 2 nodes is the farthest common ancestry those nodes
- Exists only for a certain root of the tree
- Can change for different roots

LCA results

- $dist(A, B) = level_A + Level_B - 2 * Level_{LCA}$



Lowest Common Ancestor for **Node 9** and **Node 11** is **Node 2**

How to find LCA?

- A powerful technique called binary lifting is used
- Crux of the idea is the base 2 representation of any number
- Move in incremental steps of powers of 2
- Time Complexity : O(log n)
- Space Complexity : O(n log n)

```cpp
const int N = 4e4 + 2;
const int LN = 20;

vector < int > g[N];
int tin[N];
int tout[N];
int timer;
int up[N][LN];
int m;

void dfs (int u, int p) {
  tin[u] = ++timer;
  up[u][0] = p;
  for (int i = 1; i < LN; ++i) {
    up[u][i] = up[up[u][i - 1]][i - 1];
  }
  for (auto &to : g[u]) {
    if (to != p) {
      dfs(to, u);
    }
  }
  tout[u] = ++timer;
}

bool is_ancestor (int u, int v) {
  return tin[u] <= tin[v] && tout[v] <= tout[u];
}

int lca (int u, int v) {
  if (is_ancestor(u, v)) return u;
  if (is_ancestor(v, u)) return v;
  for (int i = LN - 1; i >= 0; --i) {
    if (!is_ancestor(up[u][i], v)) {
      u = up[u][i];
    }
  }
  return up[u][0];
}
```

LCA - I

- Find the maximum in path between 2 nodes for a rooted tree

K^{th} ancestor

- Main idea : Represent the number k in binary
- Move in powers of 2 using the table built during binary lifting

```
30  const int N = 4e4 + 2;
29  const int LN = 20;
28
27  vector < int > g[N];
26  int tin[N];
25  int tout[N];
24  int timer;
23  int up[N][LN];
22  int m;
21
20  void dfs (int u, int p) {
19    tin[u] = ++timer;
18    up[u][0] = p;
17    for (int i = 1; i < LN; ++i) {
16      up[u][i] = up[up[u][i - 1]][i - 1];
15    }
14    for (auto &to : g[u]) {
13      if (to != p) {
12        dfs(to, u);
11      }
10    }
 9    tout[u] = ++timer;
 8  }
 7
 6  int kth_ancestor (int u, int k) {
 5    for (int i = 0; i < LN; ++i) {
 4      if (k >> i & 1) {
 3        u = up[u][i];
 2      }
 1    }
31    return u;
 1  }
```

Euler Tour

- A way to make an array representation of the tree
- Useful for answering queries regarding subtrees
- Many ways to represent Euler Tour
- Will study the one derived from what we learnt with the in-time/out-time concept

Training Tasks

1. https://codeforces.com/contest/208/problem/E
2. https://www.codechef.com/submit/LGSEG?tab=statement