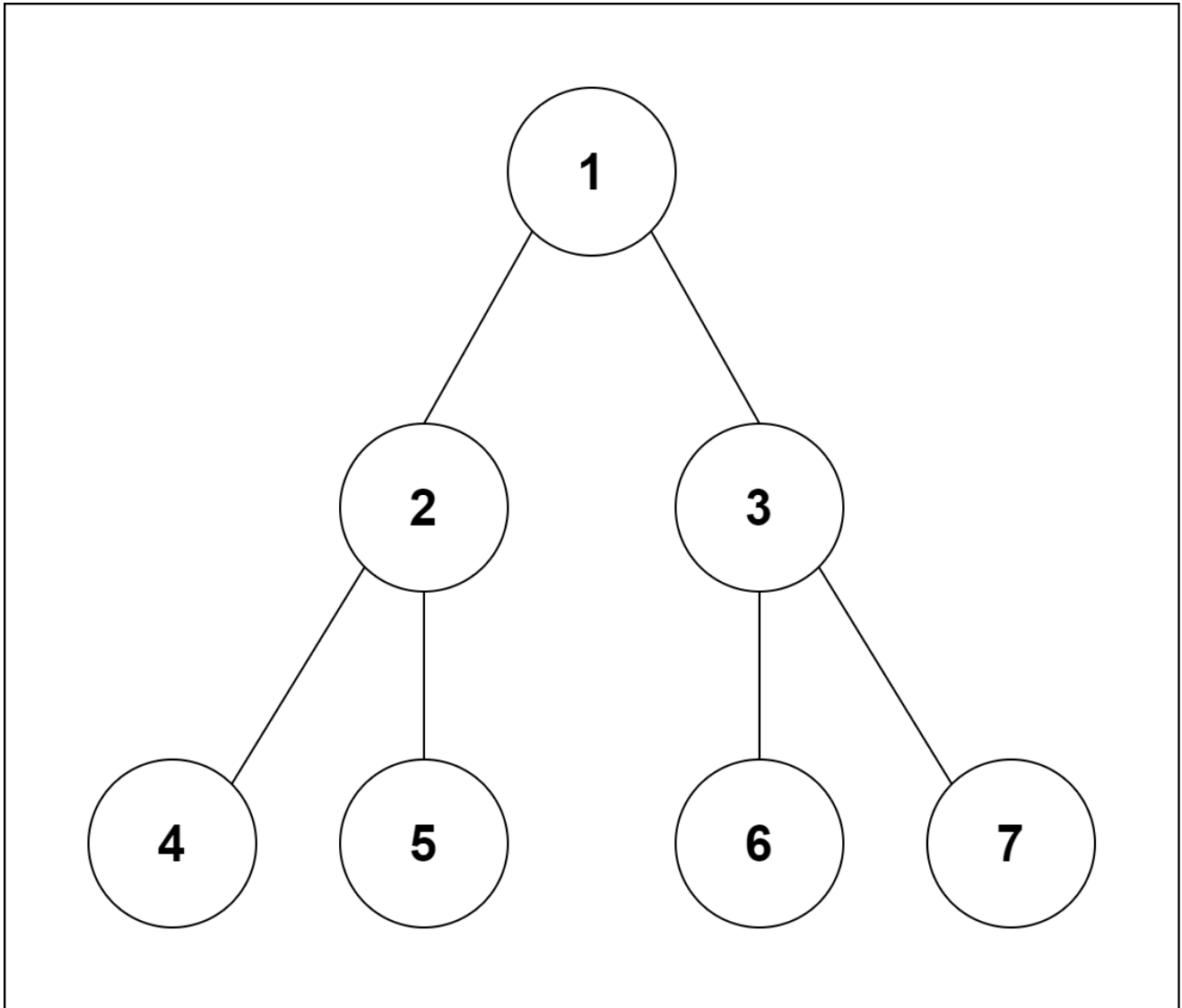


Perfect Binary Tree

A perfect binary tree is a special kind of tree. Here's what makes it special:

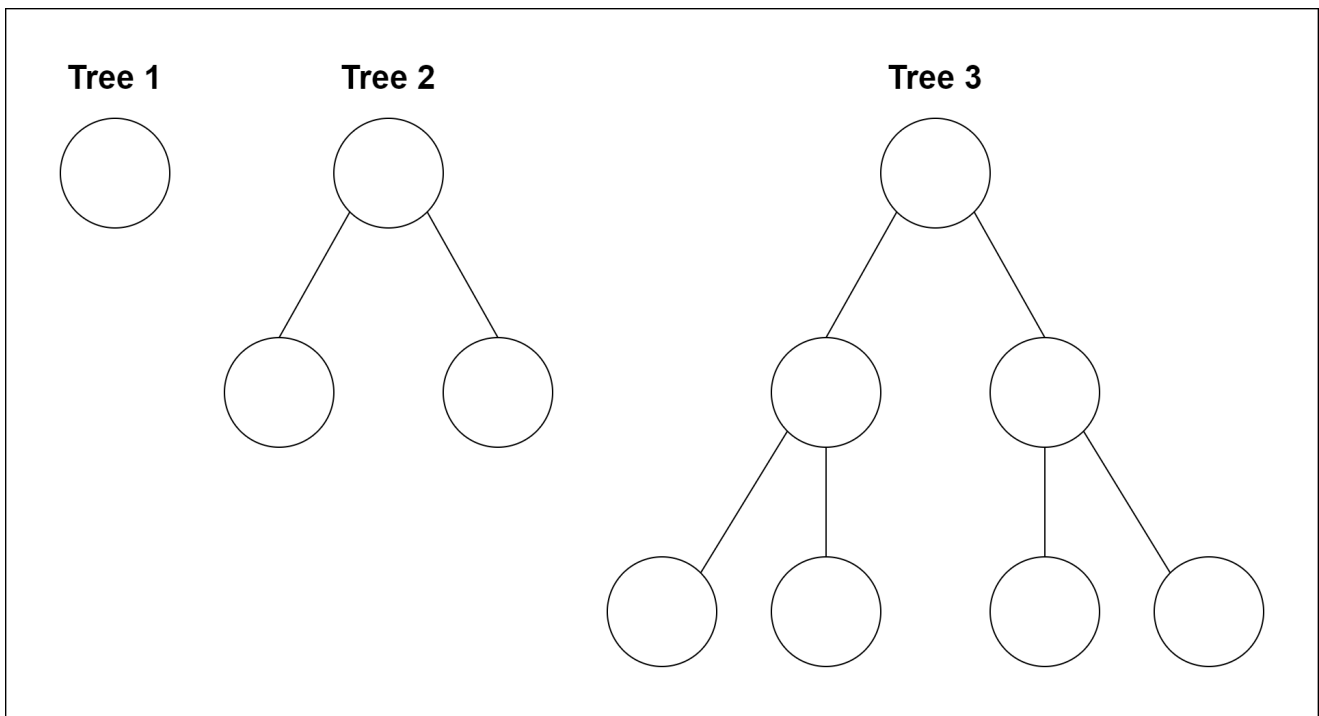
- In this tree, every node inside it has exactly two children. Not one child, not three, just two.
- Another thing is, all the leaves, which are the nodes at the very bottom of the tree, they're all at the same level. That means they're all the same distance from the top.



We can also explain a perfect binary tree in a more mathematical way:

1. If there's only one node, and it doesn't have any children, we call that a perfect binary tree of height 0.
2. If there's a node and it has some height, let's say " h ", then it's a perfect binary tree if both of its children are also perfect binary trees, but they have a height of " $h - 1$ ". And one more thing, they don't overlap; they are separate.

So, basically, a perfect binary tree is like a tree where every node has two children, and all the leaves are on the same level.



Perfect Binary Tree Theorems

1. A perfect binary tree of height h has $2^{h+1} - 1$ nodes.
2. A perfect binary tree with n nodes has height $\log(n + 1) - 1 = \theta(\ln(n))$.
3. A perfect binary tree of height h has 2^h leaf nodes.
4. The average depth of a node in a perfect binary tree is $\theta(\ln(n))$.

C++ Example

```
// Checking if a binary tree is a perfect binary tree in C++

#include <iostream>
using namespace std;

struct Node {
    int key;
    struct Node *left, *right;
};

int depth(Node *node) {
    int d = 0;
```

```

while (node != NULL) {
    d++;
    node = node->left;
}
return d;
}

bool isPerfectR(struct Node *root, int d, int level = 0) {
    if (root == NULL)
        return true;

    if (root->left == NULL && root->right == NULL)
        return (d == level + 1);

    if (root->left == NULL || root->right == NULL)
        return false;

    return isPerfectR(root->left, d, level + 1) &&
        isPerfectR(root->right, d, level + 1);
}

bool isPerfect(Node *root) {
    int d = depth(root);
    return isPerfectR(root, d);
}

struct Node *newNode(int k) {
    struct Node *node = new Node;
    node->key = k;
    node->right = node->left = NULL;
    return node;
}

int main() {
    struct Node *root = NULL;
    root = newNode(1);
    root->left = newNode(2);
    root->right = newNode(3);
    root->left->left = newNode(4);
    root->left->right = newNode(5);
    root->right->left = newNode(6);

    if (isPerfect(root))
        cout << "The tree is a perfect binary tree\n";
    else
        cout << "The tree is not a perfect binary tree\n";
}

```