

1. Problem analysis

This project needs to find all the paths from the root node to every leaf node with the same given sum of the weights, and output the weight of each node along each the path in order.

Thus this problem has been turned into two parts, one for going through all the path from the root node to the leaf node, the other for outputting the weights sequence in order.

And our algorithm can be divided into two parts, the depth-first-search algorithm and quicksort algorithm.

2. Algorithm analysis

Using Depth-first-search algorithm to find out all the paths from the root node to the leaf node with the given sum of the weights, then print them out after sorting.

2.1 Data structure

Record all the nodes in an array of structure

```
struct TreeNode
{
    long Weight;
    vector<int> Children;
};
vector<TreeNode> Tree;
```

Record all the paths in a two-dimension array

```
vector<vector<long>>> Path;
```

2.2 Function analysis

2.2.1 Depth-First-Search for each node

```
Function AddNode(node_a, weight, stack)
{
    If node_a is an leaf node and weight+node_a.weight = the given sum of the weights then
        Push weight into the stack
        Path = all the elements in the stack
        Return
    Else
        For all the children of node_a
            Push child.weight into the stack
            Addnode(child, weight, stack)
            Pop out child in the stack
        End for
    End if
}
```

2.2.2 Quick sort using C++ standard library function

```
void sort (RandomAccessIterator first, RandomAccessIterator last, Compare comp);
sort(Path.begin(), Path.end(), greater<vector<long>>())
```