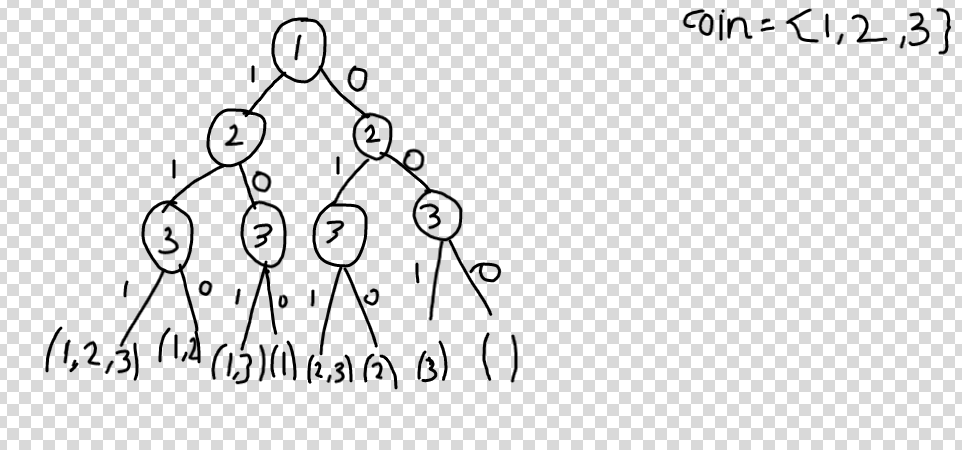
**Problem :** [**https://cses.fi/problemset/task/1745/**](https://cses.fi/problemset/task/1745/)

**Approach :**



-> Either take a coin or not take it.

-> All the leaves give the sum of that particular subset.

-> This way there are 2^n leaves, for 100 coins its 2^100 leaves and it will give TLE.

Max value of coin is 1000 so max sum of all coins is 10^5 .

-> In short , when we reach a particular coin valuation, we would already have a particular sum passed to it, e.g in leftmost path in above tree, sum=1+2=3 is passed to 3, so if the same sum=3 is passed to 3 in any other subtree , we need not visit that subtree at all .

-> So store dp[3][3]=true, so when it’s called again we won’t visit that subtree

**->Any sum value from 0 to 10^5 can be passed to a particular node,so in worst case, for each of 100 coins ,sum values from 0 to 10^5 need to be computed in tree, and each value is visited only once as we are using DP,so array dp[100][100000] is needed and time = O(no of coins\*sum of all given coins)**

**Code :** [**https://cses.fi/problemset/result/2467360/**](https://cses.fi/problemset/result/2467360/)