**Problem :** [**Coin Combinations I**](https://cses.fi/problemset/task/1635/)

**Approach : TREE Brute force.**

-> We need to make a sum ‘X’ using ‘n’ given coins on some value and INFINITE coins of these values are available.

-> So when we want a particular sum, we recursively check if (sum-coin[i]) can be made using all the coins and that will be similarly calculated by the tree below, and if any leaf returns YES , then that is a Valid Path.

-> If for particular sum , we already checked how many combinations are available, then we store it in dp[sum].

-> So bcoz of dp , At most there can be ‘X’ sums we need to calculate in the Tree (if one of the coins is of value=1) , and for each sum we need to check ‘n’ coins(means if a value of sum-coin[i] is possible or not), for example for bottommost level sums we check n coins , then the answer of that is returned above, and there too ‘n’ coins are checked,using the answer calculated below.

**-> So time = (N\*X)**

**Code :** [**https://ideone.com/KNmb33**](https://ideone.com/KNmb33)

**Problem :** [**Coin Combinations II**](https://cses.fi/problemset/task/1636)

**Approach :** [**Dynamic Programming : Coin Combinations**](https://youtu.be/-pXjopzMVrE)

-> We start from taking last coin till the first.

-> At every instance you have 2 choices ,

1. **INCLUDE** the current coin and search if a sum of “remainingSum - currCoinValue” is possible in how many paths in the tree formed below ,and bcoz we included the current coin, that is why in the below tree we are still permitted to use the current coin .
2. **DON’T** include the current coin, and check whether “remainingSum” is possible in how many paths in the tree formed below , and **once we excluded the current coin,we can’t use it in any tree below.**

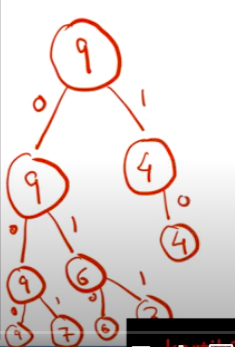
**->** So the Basic Intuition is that first we try to include a coin different number of times and then exclude that coin completely from rest of that path.

So if coins are 2,5 and sum=14

Then if we found a path 5+5+2+2 = 14

We will never get something like 5+2+5+2 bcoz once we reached 2 , we can’t go back to 5 so only 1 such permutation is recorded not all.

E.g : coins = 2,3,5 and sum=9 , a short tree looks like below.



So formula : dp[i][sum] = dp[i-1][sum] + dp[i][sum-valueOf ith coin]

=> Means check in how many paths , a current sum of “sum” is possible from rest ‘i-1’ coins +

Include the current coin and check , in how many paths , a sum of “sum-valueOf ith coin” is possible from ‘i’ coins.

**Code :** [**https://ideone.com/hdzLxE**](https://ideone.com/hdzLxE)