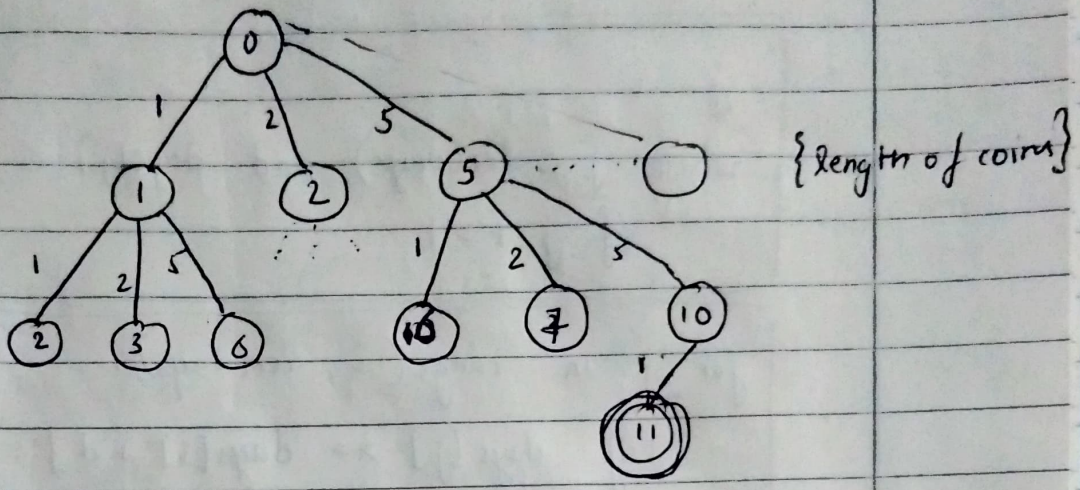
322 - Coin Change Problem

Start with brute force:- Decision tree is best to understand

Ex:- coins = [1, 2, 5] , amount = 11



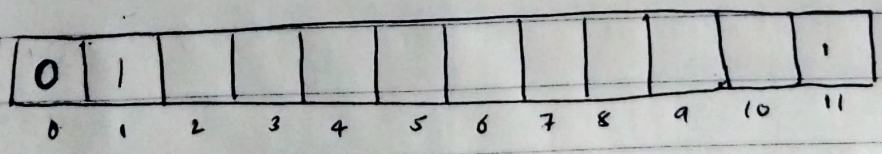
In the above brute force approach the time complexity is exponential

worst case complexity : $length^{amount}$ $\therefore (b^a)$

But with **memorization** we can stop the repeated calculations.

Tabulation method can be used as well:

DP =



As seen in memorization decision tree, we are adding {1, 2, and 5 at each step to reach the target.

Here we will build bottom up DP[i] to make the next.

$$\forall i \in [0, 11]$$
$$\forall j \in \text{~~costs~~ coins}$$
$$DP[i] = \{ \min(DP[i], DP[i - \overset{j}{\text{~~coins[j]~~}}] + 1) \mid i >= coins[j] \}$$

In the tabulation steps:

We derived the mathematical formula that can be translated to code:

Example to understand

DP[2] = ~~for j+1~~ float('inf')

for j = 1 $\quad DP[2] = \min(DP[2], DP[2-1]+1)$

This means we have used 1 coin of ①
and what's left now amount = 1,
that can be done by DP[amount]

$DP[2] = \min('inf', 1+1) = 2$

for j = 2 $\quad DP[2] = \min(DP[2], DP[2-2]+1)$

$= \min(2, 0+1) = 1$

Using this we can reach DP[1]