Date : 03/07/19
Page No.

~~min coin required prob~~

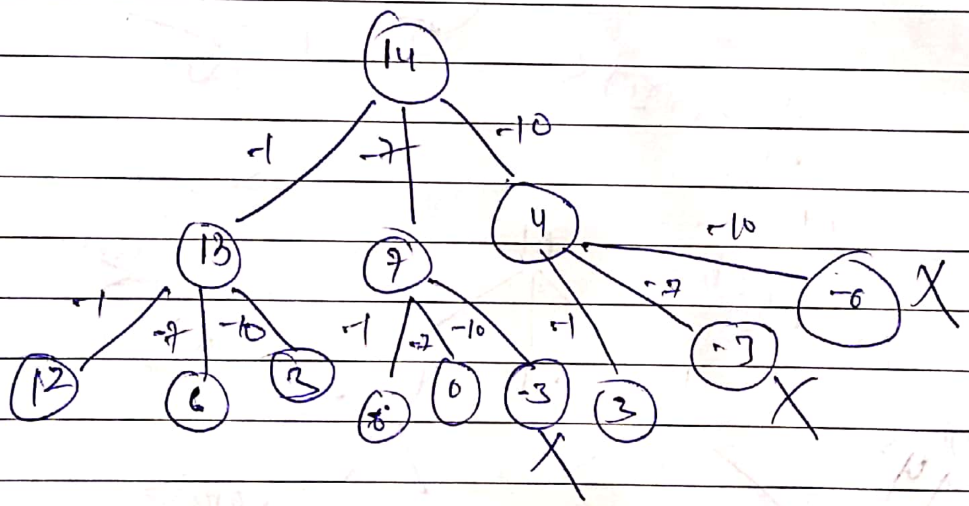
Currency Exchange Problem :-

amount = 14   coins = {1, 7, 10}

prob : min coins required to sum up the amount ?

① Recursion Approach (Brute Force) :-

try every possible coins (start with)



pseudo code:

```
                min coin needed (amount, coins [ ]) {
                        if (amount == 0) return 0;
                        if (amoun < 0) return INT_MAX;
min_coins = ~~vector<int> dp~~ ( ~~amount~~, INT-MAX) ;
INT_MAX;            ~~dp [0]~~ = 0 ;
                for (i = 1; i <= amount; i++) {
                        int tmp = 1 + min coinneeded (amount - i, coins);
```

if (tmp < ~~INT_MAX~~ mincoins){

dp[i] = ~~tmp~~ ; min_coins = tmp ;

}

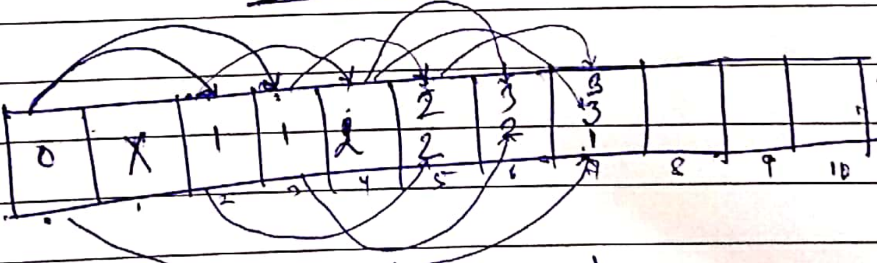
}

return ~~dp [amount]~~ min_coins ;

(2) dp approach

In dp approach we have to see the total no cells ~~we~~ from where when we can reach to for any cell i.e: i th there could be maximum of K th ways where K is no. of coins. And total coins needed to ~~reach~~ reach ith coin to make i amount

tmp = 1 + coins needed (i - K_j) j = 0,1,2,3 - size of coins array



coins 2, 3, 7

take minimum of each.

```
min coinstopdowndp (amount, coins) {
vector<int> dp (amount+1, INT_MAX);
dp [0] = 1;
 for ( i=1; i<= amount ; i++) {
            for ( j=0; j< coins.size(); j++) {
                 if (i - coins [j] >= 0 & dp [i - coins [j]] != INT_MAX)
                        dp [i] = min(dp [i], 1+ dp [i-coins[j]]);
     }}
 return dp [amount];
}
```