Q! Coin change (unordered)

n different denominations

no of ways to get total = k

One coin can be used multible times

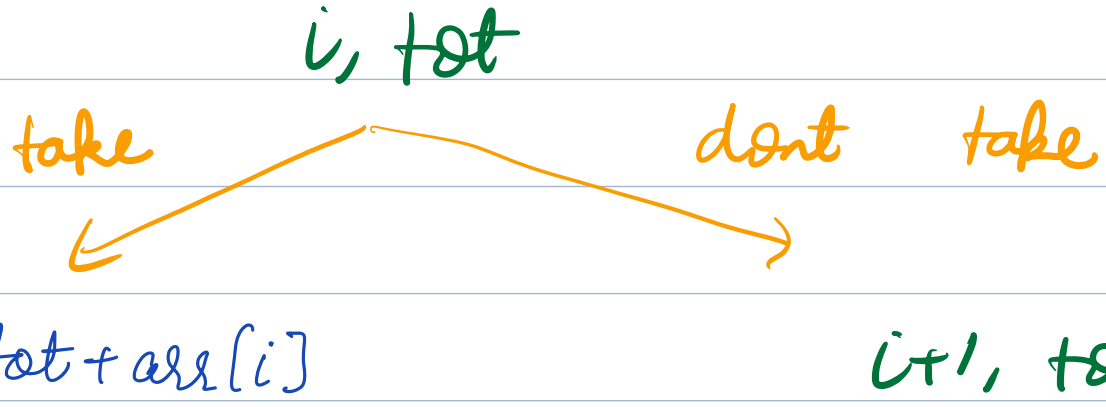
Eg 7 3

k = 10 ans = 1

Brute force: Try every posible combinations. TC: exponential

i → index of coin

tot → total sum till now




dp(i,j) = dp(i+1, j) + dp( i, j + arr[i] )

Code

```
dp[n][k+1]        // =-1
int calc( int i, int j){
  if(i==n){
     if (j==W)   return 1
     else        return 0
  }
  if( j>W )
        return 0
if( dp[i][j] != -1 )
        return dp[i][j]

ans =      calc(i+1,j)   + calc(i, j+wt_i)
dp[i][j] = ans
return ans
}
```

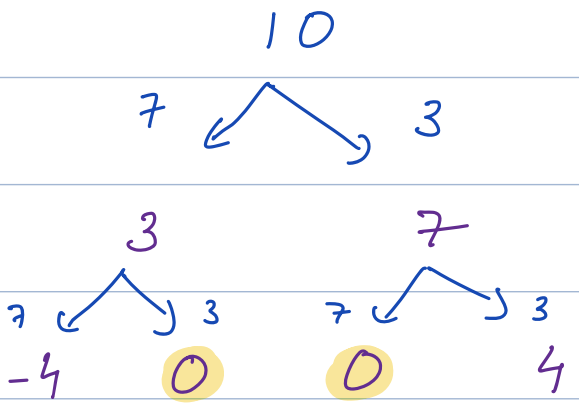
TC } O(NW)
SC }

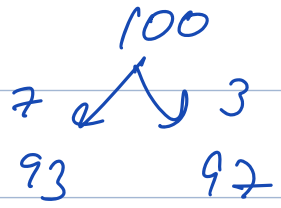
Q2 Coin change ordered ( you can take any coin multiple times )

7 3 k = 10

{7,3} {3,7} ans = 2

ans(100) = ans(93) + ans(97)




dp(tot) = dp(tot - $a_0$) + dp(tot - $a_1$) + . . . .

. . . + dp(tot - $a_{n-1}$)

Code

```
int dp[k+1]
int ways(int tot){
    if(tot<0)
        return 0
    if(tot==0)
        return 1
    if(dp[tot]!=-1)
        return dp[tot]
    ans = 0
    for(i=0;i<n;i++){
        ans += ways(tot - coins[i])
    }
    dp[tot] = ans
    return ans
}
```

TC: $O(nk)$

SC: $O(k)$

idx, weight → val

idx, val → weight

Q5 0-1 Knapsack problem

weight is constrained, find max possible value

$1 \le N \le 500$

$1 \le value[i] \le 50$

$1 \le weight[i] \le 10^9$

$1 \le W \le 10^9$

dp[N][W]

$500 \times 10^9$

$5 \times 10^{11}$

Can you do the original soln ?

i, weight

500 $10^9$

$\rightarrow$ $500 \times 10^9$

$5 \times 10^{11}$

not possible

Think other way around.

i, total_val $\rightarrow$ min weight

take

weight[i] +

i+1, tot_val + value[i]

dont take

i+1, total_val

$1 \le N \le 500$

$1 \le value[i] \le 50$

$tot\_val = 500 \times 50 = 25,000$

$N \times tot\_val$

$500 \times 25,000 = 125 \times 10^5$

$= 1025 \times 10^7$

Think other way around.

i, total_val till now ⇒ min weight

weight[i] + i+1, tot_val + value[i]

i+1, total_val

| | | | | | |
|---|---|---|---|---|---|
| Eg | val | 10 | 40 | 20 | 30 |
| | weights | 500 | 100 | 700 | 600 |

ans (0, 50) = 600

ans (0, 30) = 600 // take 30 only

ans (0, 15) = ∞

ans (0, 0) = 0

ans (0, 1) = 10

ans (0, 2) = 5

: = 12

:

:

20k

ans (0, 25000) => ?

capacity = 11

{ans}

abab

bbaa

aca

Uber, Intuit, Scaler

## Q Rod cutting

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| A = { | 1 | 4 | 2 | 5 | 6 } |

N = 5

| sold len | profit |
|---|---|
| 5 | 6 |
| 4+1 | 6 |
| 3+2 | 6 |
| 3+1+1 | 4 |
| 2+2+1 | 9 |
| . | . |
| . | . |

| | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| A = { | 1 | 4 | 2 | 5 | 6 } | ⇐ value |
| | | | | | | ⇐ weight |

0-N knapsack

total = N

knapsack capacity = N

TC } O(NW)
SC } = O(N²)

aca ⇒ 2

aaa

doubt

abb

acb

⇒ 3

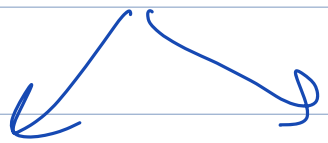
idx, prev char

abcde

aaddd

idx

dp(n)(26)



accde

↓

accce