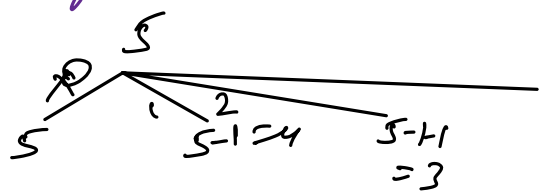
Welcome ☺

Agenda : Knapsack.
2 - 4 questions.

Q Given a rod of length N and an array A of length N

A[i] ⇒ price of i length rod (index is 1 based)

Find max value that can be obtained by cutting rod into some pieces & selling them.



eg: N = 5

A = [1 4 2 5 6]
1 2 3 4 5

sold	length	prices.
5		6
1 + 4		6
3 + 2		6
3 + 1 + 1		4
2 + 2 + 1		9 ←
2 + 1 + 1 + 1		7
1 + 1 + 1 + 1 + 1		5

Unbounded Knapsack.

State ⇒ length of rod.

W → length of rod.
wt[i] → length of piece of rod.
value[i] → A[i]

dp[i] ⇒ max value if length is i

∀i dp[i] = 0

```
for( i = 1 → N )   // capacity / length.
{
   for( j → 1 to i )
   {
      dp[i] = max( dp[i], A[j] + dp[i-j] )
   }
}
return dp[N]
```

Q In how many ways can we make N by using coins given in the array.

A[i] ⇒ value of i^{th} coin

One coin can be used multiple times

A: Ordered selection (x, y) ≠ (y, x)

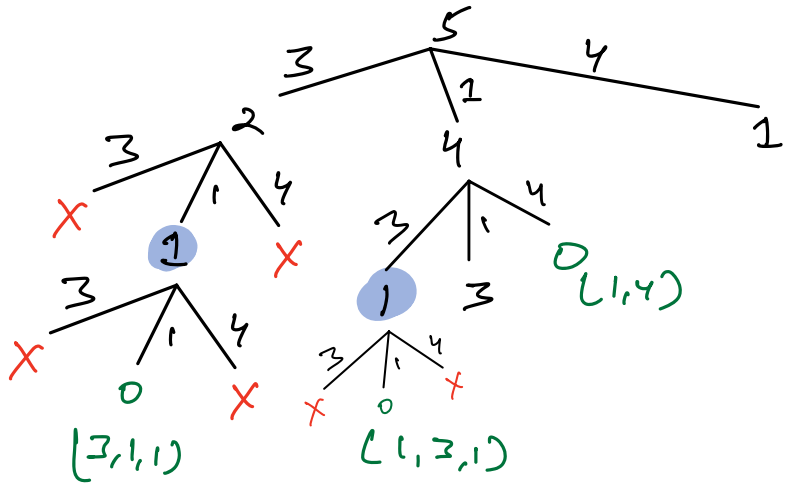
eg: N = 5

A: [3 1 4]

(1,4) { 3, 1, 1 } { 1, 1, 3 }

(4,1) { 1, 3, 1 } { 1, 1, 1, 1, 1 }

Ans = 6



dp[i] ⇒ # ways to select coins s.t sum = i

∀i dp[i] = 0

dp[0] = 1

for (i → 1 to N)

{

for (j → 0 to A.length() - 1)

{

if (A[j] ≤ i)

{

dp[i] = dp[i] + dp[i - A[j]]

}

}

}

return dp[N]

T.C = O(N * A.length())

S.C = O(N)

Q: Un-ordered

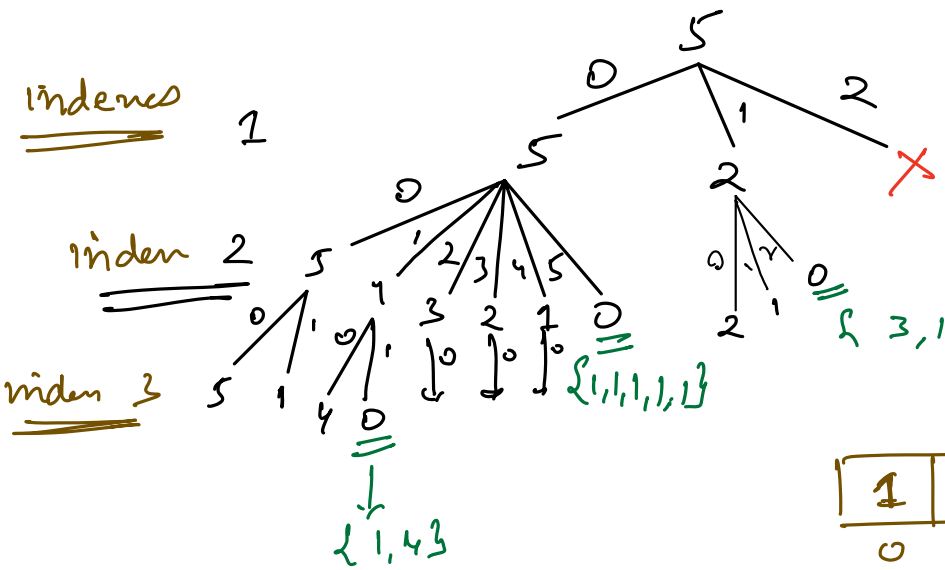
$(x, y) = (y, x)$

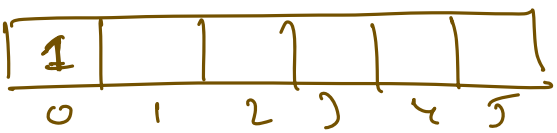
eg: N = 5

A: [3 1 4]
1 2 3

(1,4) {3,1,1} {1,1,3}
(4,1) {1,3,1} {1,1,1,1,1}

Ans = ~~6~~ 3





dp[i] ⇒ # ways to select coins s.t sum = i

j = 0
coin ⇒ 3

```
∀i dp[i] = 0
dp[0] = 1
for ( j → 0 to A.length()-1 )   // inden
{
    for ( i → 1 to N )   // sum
    {
        if ( A[j] ≤ i )
        {
            dp[i] = dp[i] + dp[i - A[j]]
        }
    }
}
return dp[N]
```

T.C = O(N * A.length())
S.C = O(N)

Q We are given N toys with their happiness & weight. Find max. total happiness that can kept in a bag. with capacity W. 0-1

$0 \leq N \leq 500$

$0 \leq h[i] \leq 50$

$0 \leq wt[i] \leq 10^9$

$0 \leq W \leq 10^9$

T.C => $O(N * W)$

$500 * 10^9 = 10^{11}$

TLE

Change state