Q → Given a rod of length N & an array A of length N.

A[i] → price of (i+1) length rod.

Find max profit we can earn by cutting the rod & selling them.

$N = 5$

$$\begin{array}{rccccc} & 0 & 1 & 2 & 3 & 4 \\ A = & [1 & 4 & 2 & 5 & 6] \\ \text{len} \rightarrow & 1 & 2 & 3 & 4 & 5 \end{array}$$

All possibilities $= \underline{2^{N-1}}$

Parts of rod
1) length ✓
2) A[i] i.e value ✓
Total length of rod = N

Unbounded Knapsack

Sold length	Total Value
5	6
4 + 1	5+1 = 6
3 + 2	2 + 4 = 6
3 + 1 + 1	2 + 1 + 1 = 4
2+2+1	4 + 4 + 1 = 9 (Ans)
2+1 +1+1	4 + 1 + 1 + 1 = 7
1+1+1+1+1	1 + 1 + 1 + 1 + 1 = 5

dp[i] → max value that can be achieved via rod of length 'i'.

```
dp[0] = 0        ∀i, dp[i] = 0
for i → 1 to N {    // length of rod to be sold
    for j → 1 to i {    // length of part
        dp[i] = max (dp[i], A[j-1] + dp[i-j])
    }
} return dp[N]
```

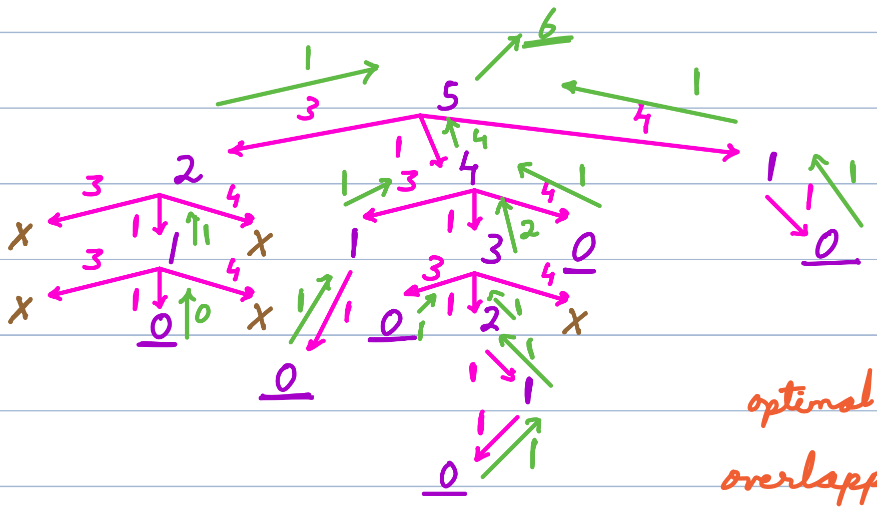
TC = $\underline{O(N^2)}$ SC = $\underline{O(N)}$

Q → In how many ways can we get sum = N using the coins present in the array.
A coin can be selected multiple times.

A → Ordered selection of coins $(x, y) \neq (y, x)$

N = 5   {1, 4} {3, 1, 1} {1, 1, 3}

A = [3 1 4]   {4, 1} {1, 3, 1} {1, 1, 1, 1, 1}   Ans = 6



optimal substructure ✓

overlapping subproblems ✓

DP

dp[i] → # ways to get sum 'i'

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

TC = O(N * A.length)

SC = O(N)

B → Unordered Selection of Coins $(x, y) = (y, x)$

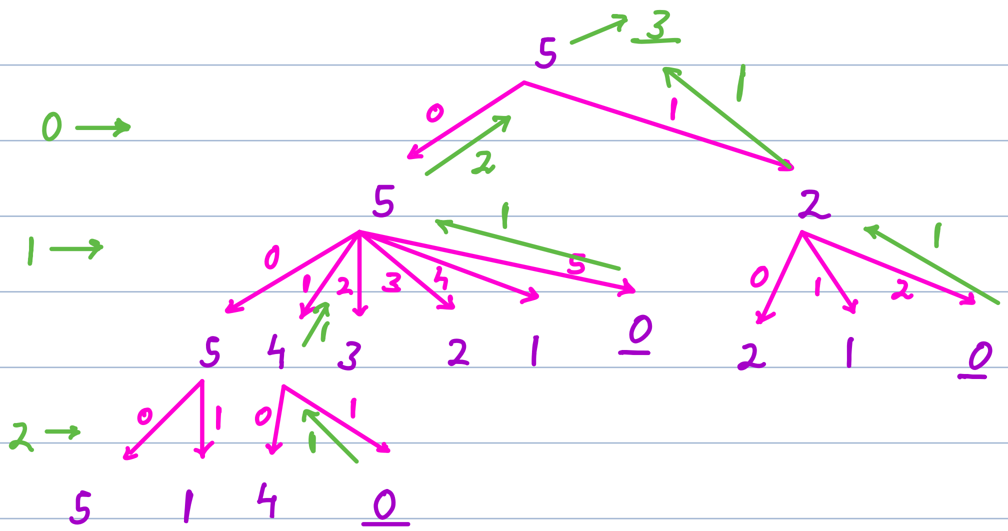
$N = 5$

$A = [3 \quad 1 \quad 4]$ (indices 0 1 2)

{1, 4} {3, 1, 1} {1, 1, 3}

{4, 1} {1, 3, 1} {1, 1, 1, 1, 1}

Ans = 3



dp[i] → # ways to get sum 'i'

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

$TC = O(A.length * N)$

$SC = O(N)$

$$\begin{tikzcd}[row sep=small] & \text{happiness} \\ Q \rightarrow N \text{ toys} \arrow[ur] \arrow[dr] & \\ & \text{weight} \end{tikzcd}$$

Find max total happiness that can be kept in a bag with capacity W. 0-1 Knapsack

TC = O(N * W)

constraints

$1 <= N <= 500$

$1 <= h[i] <= 50$

$1 <= w[i] <= 10^9$

$1 <= W <= \bigcirc\!\!\!\!\!\!10^9$

dp[index][capacity] → max profit

max total $H = 50 * 500$

$= 25000$

dp[i][j] → min weight req. to get happiness 'j' considering first 'i' elements.

$500 * 25000$

$= 1.25 * 10^7$

$$dp[i][j] \begin{cases} \xrightarrow{\times} dp[i-1][j] \\ \xrightarrow{\checkmark} wt[i] + dp[i-1][j-h[i]] \end{cases} \Big\} \min$$

```
for i → 0 to N {    // elements 1 → N
    for j → 0 to (50 * N) {
        if (i==0 || j==0)   dp[i][j] = 0
        else if (h[i] <= j) {
            dp[i][j] = min(dp[i-1][j], wt[j]+
                                       dp[i-1][j-h[i]])
        } else { dp[i][j] = dp[i-1][j]
        }
}
}
for i → H to 0 {
|    if ( dp[N][i] <= W)
|        return i
}
```

$TC = O(N * 50 * N) \rightarrow \underline{O(N^2)}$

$SC = O(N * 50 * N) \rightarrow \underline{O(N^2)}$