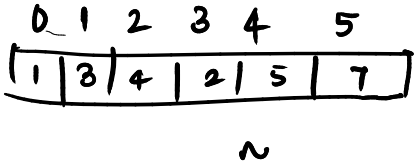
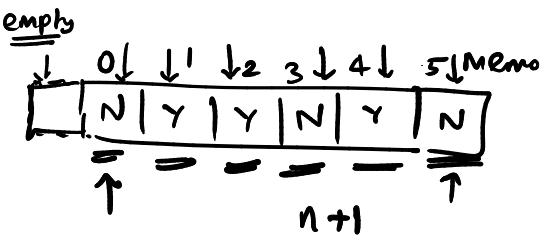
Crude example :





Recursion : (_ _)

// Base case.

when I have nothing to pick / no items are left more :

return 0

Memo.

Recall → lookup memo

Recursion → (mechanical process) → DP.

| Do the work.

memo[n] = ans. ← storing the ans in memo.

return ans

Recap : Knapsack → Template.

↳ variations → 1. Subset sum

2. Equal sum partition */

3. min subset sum difference

4. Target sum. → ✓ leetcode

2: Unbounded knapsack.

1. Either pick or skip.
2. If you pick → you have choice to pick multiple times.

↳ Coin change

coins = {1, 3, 4} → o/p : 2

Capacity / change = 6 → {3, 3}

+1 +1 +0 / 2

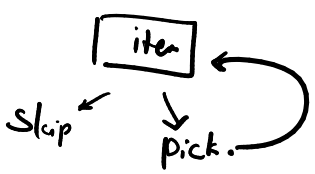
Knapsack.

```
int knapsack (int arr, int n, int C)
    if n==0 || C < 0  :   if C==0:
        return INT_MAX        return 0
                10e8
    if arr[n] > C :
        return knapsack (arr, n-1, C)
    else :
        return min { knapsack (arr, n--, C) ,
                     knapsack (arr, n/n, C-arr[n]) + 1 }
                                  n
```

Assumes: arr[i] can be picked only once.

→ Unbounded

Min coins that make up for change C.



10***