Q:| Given arr[N], find the no. of subsets with sum = K. Array elements > 0 & K > 0

$$\begin{array}{lcccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ arr[8] : \{ & 7, & 4, & 9, & 6, & 10, & 13, & 11, & 14 & \} \end{array} \quad K = 22$$

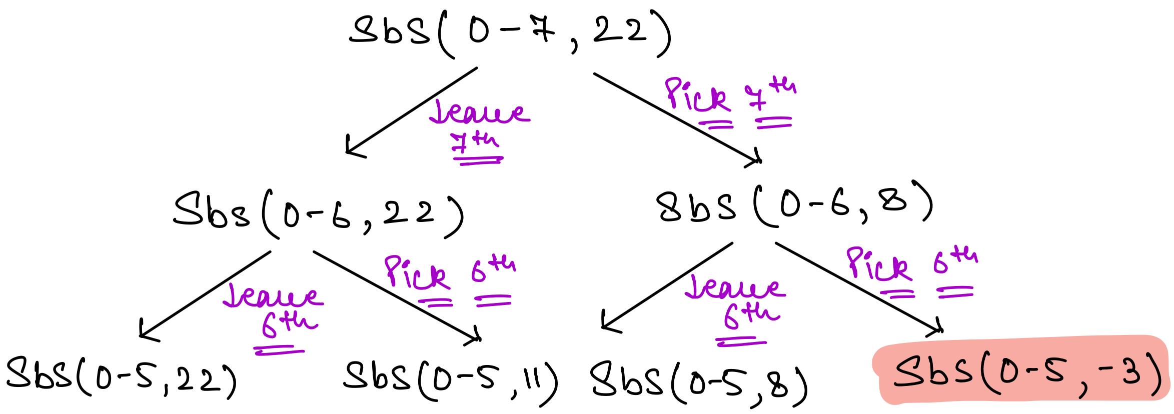
{7, 4, 11}
{9, 13}
{7, 9, 6}

③

Ideas :- Generate all possible subsets & check if sum == K.

→ Bit Masking* ⇒ TC: $O(N \cdot 2^N)$

→ Backtracking ⇒ TC: $O(2^N)$



Ex:-

K = 45

_ _ _ _ _ _	6	10	14	2	8	
0 1 2 3 4 5	6	7	8	9	10	
6 elements, S = 25	✓	*	✓	*	*	⇒ 20
6 elements, S = 25	✗	✓	✗	✓	✓	⇒ 20

$$\left.\begin{array}{l}\rightarrow \text{Overlapping Subproblems.} \\ \rightarrow \text{Optimal Substructure.}\end{array}\right\} \text{DP.}$$

DP State

$dp(i, j)$: # No. of subsets using elements from $[0, i]$ & sum $= j$

* $$dp(i,j) = \left\{ \begin{array}{ccc} \text{leave } i^{th} & & \text{pick } i^{th} \\ dp(i-1,j) & + & dp(i-1, j-a[i]) \\ & & j-a[i]>=0 \\ & & j>=a[i] \end{array} \right\}$$

* **dp Table**

No. of subsets using elements from [0 to N-1] with Sum = K

↳ ans: $dp[N-1][K]$

Size :- $dp[N][K+1]$

* **Base Case** :-

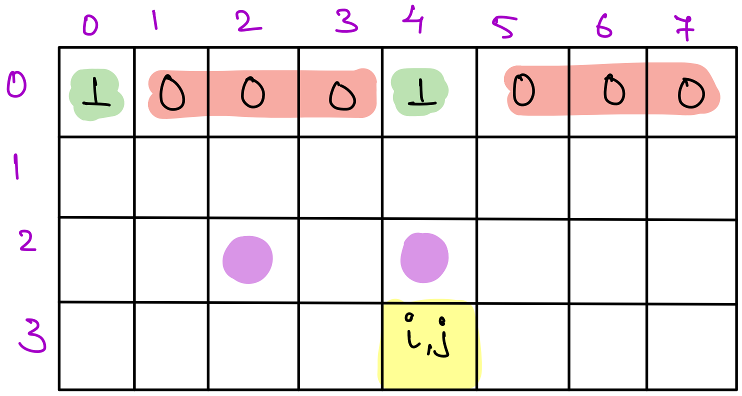
$$dp(i,j) = dp(i-1,j) + dp(i-1, \underbrace{j-A[i]}_{j>=A[i]})$$

→ $i=0$: Base case

$\quad\quad\ \ 0\quad 1\quad 2\quad 3$

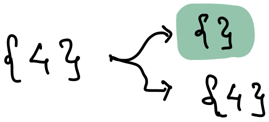
A[4] : { 4, 2, 1, 3 }   K = 7

dp[4][8]



	0	1	2	3	4	5	6	7
0	1	0	0	0	1	0	0	0
1								
2			●		●			
3					i,j			

dp[0][0] : ①
No. of subsets using indices [0 - 0] with sum = 0.



dp[0][1] = No. of subsets using indices [0 - 0] with sum = 1.

dp[0][4] : 1

```
for ( j = 0 ; j <= K ; j++ )
        dp[0][j] = 0
dp[0][0] = 1
if (A[0] <= K) { dp[0][A[0]] = 1 }
```

Base Case

* dp[1][0] : No. of subsets using indices [0 - 1] with sum = 0. ⇒ {4, 2}

{ } ⇒ sum = 0

Eg* { 4 0 1 }

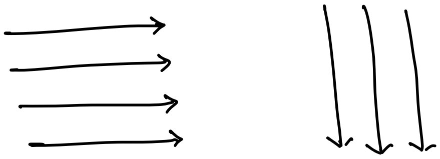
No. of subsets with sum = 0 ⇒ 2. { }, {0}

Ex { 4, 0, 1, 0 }

No. of subsets with sum = 0 ⇒ $\underline{2^2}$

{ }
{0} {0}
{0 0}

Code :-



```
for ( i = 1 ; i < N ; i++ ) {
    for ( j = 0 ; j <= k ; j++ ) {
        dp[i][j] = dp[i-1][j]
        if ( j >= a[i] )
            dp[i][j] += dp[i-1][j-a[i]]
    }
}
return dp[N-1][K]
```

TC : O(N·K) , SC : O(N·K)

TODO

A[4] : { 4, 2, 1, 3 }   K = 7

(indices: 0, 1, 2, 3)

dp[4][8]

	0	1	2	3	4	5	6	7
0	1	0	0	0	1	0	0	0
1								
2								
3								

* Space Optimization

$dp[2][K]$: TC : $O(N \cdot K)$, SC : $\underline{\underline{O(K)}}$

Q: Given N array elements, find the length of smallest subset with sum = K.

$$arr[8] : \{ \overset{0}{7}, \overset{1}{4}, \overset{2}{9}, \overset{3}{6}, \overset{4}{10}, \overset{5}{13}, \overset{6}{11}, \overset{7}{14} \} \quad K = 22$$

$$\Rightarrow \{13, 9\} \Rightarrow \underline{\underline{2}}$$

$dp[i,j]$: length of smallest subset with sum = j using the elements from [0-i]

$$dp[i,j] : \underline{\underline{\min}} \begin{cases} \underset{\text{leave ith}}{dp[i-1, j]}, \; \underset{\text{pick ith}}{dp[i-1][\underbrace{\overset{\text{ith index}}{j} - A[i]}_{\underline{\underline{j >= A[i]}}}] + 1} \end{cases}$$

* DP table.

$dp[N][K+1]$

* Base Case :- $\underline{\underline{i == 0}}$

Eg   A[4]: { 4 3 2 6 }   K = 7

dp[4][8]

	0	1	2	3	4	5	6	7
0	0	5	5	5	1	5	5	5
1								2
2								
3								

dp[0][0] : 0

length of the smallest subset with sum = 0 from [0-0].

$$\begin{tikzcd} & \{\} \ \text{len} = 0. \\ \{4\} \arrow[ur] \arrow[dr] & \\ & \{4\} \end{tikzcd}$$

dp[0][1] : length of the smallest subset with sum = 1 from [0-0].

dp[0][4] : 1

dp[0][5] :

$$dp[1][7] = min(dp[0][7], dp[0][\underbrace{7-3}_{4}] + 1)$$

$$dp[1][7] = min(5, 6) = 5$$

*

```
for( j=0; j<=K; j++)
        dp[0][j] = N+1;   → No subset.
dp[0][0] = 0   // Empty Subset
if( a[i] <= K){
    dp[0][a[i]] = 1
}
```

```
for(i=1; i< N; i++) {
    for(j=0; j<=k; j++){
        dp[i][j] = dp[i-1][j]
        if( j >= a[i] )
            dp[i][j]= min(dp[i][j],dp[i-1][j-a[i]]+1);
    }
}
return dp[N-1][K]
```

TC : $O(N\cdot K)$

SC : $O(N\cdot K) \longrightarrow O(K)$ $\{dp[2][K+1]\}$

Q: Check if there exist a subset with sum=K.

dp[i][j] : If there is a subset with sum = j using elements [0-i].

$$dp[i][j] : \quad dp[i-1][j] \;||\; dp[i-1][\underbrace{j-A[i]}_{j>=A[i]}]$$

DP Table : dp[N][K+1]

Base Case i==0.

A[4]: { 4 3 2 6 } K = 7

dp[4][8]

	0	1	2	3	4	5	6	7
0	T	F	F	F	T	F	F	F
1	T							
2	T							
3	T							

dp[0][0] : {4} < {} sum=0 / {4}

dp[0][1] : F

dp[0][4] =

```
for(j = 0; j <= K; j++) {
        dp[0][j] = false;
  dp[0][0] = true;
  if(a[i] <= K) {
      dp[0][a[i]] = true;
  }
```

$$\boxed{\begin{array}{l} TC: O(N \cdot K) \\ SC: O(N \cdot K) \end{array}}$$

→ dp[2][K+1]

SC: $\underline{\underline{O(K)}}$

* * *