Complexidade Subset Sum Dinâmico

Na notação big O as atribuições recebem valores constantes que serão descartados na incidência de n's

```
SUBSET-SUM-PROG-DIN (p, n, c)
1   para i crescendo de 0 até n faça                    → n
2       t[i,0] ← 1
3   para b crescendo de 1 até c faça                    → n
4       t[0,b] ← 0
5       para i crescendo de 1 até n faça                → n
6           s ← t[i−1,b]
7           se s = 0 e p_i ≤ b                          → (n -1)
8               então s ← t[i−1,b−p_i]
9           t[i,b] ← s
10  devolva t[n,c]
```

Nesse cenário, sabemos que tem dois for aninhados, o de cima varia em **c** e o de baixo varia em **n**. Se o **n** for um valor diferente, a complexidade dele é O(c*n*), *porque ele depende de dois fatores de entrada para saber a complexidade.* Se por algum propósito do algoritmo o **n** for igual ao **c**, então teremos uma complexidade O(n^2) que é o pior algoritmo que temos, porque ele cresce de forma exponencial.

Complexidade: **O(c*n)**