

Teste de Mesa - Problema da Mochila (Knapsack 0/1)

Este documento apresenta um teste de mesa detalhado para o algoritmo do Problema da Mochila 0/1, tendo como objetivo entender melhor o último exemplo de C++ apresentado na aula do Lab 03. O objetivo é determinar o maior valor possível que pode ser carregado em uma mochila com uma capacidade máxima dada.

Exemplo Utilizado:

Número de itens (N) = 3

Capacidade máxima da mochila = 5

Pesos dos itens: [2, 3, 4]

Valores dos itens: [4, 5, 6]

Inicialização da Matriz dp:

A matriz $dp[N+1][capacidade+1]$ é inicializada com zeros. Cada célula $dp[i][w]$ representa o maior valor possível que pode ser carregado usando os primeiros 'i' itens e uma capacidade máxima de 'w'.

Matriz Inicial dp:

i \ w	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0

Passo a Passo do Preenchimento da Matriz dp:

Iteração 1: Item com Peso 2 e Valor 4

Neste passo, verificamos se o item 1 com peso 2 pode ser incluído na mochila para cada capacidade **w**. Se couber, temos duas opções:

- Não incluir o item: O valor total na mochila permanece o mesmo que na iteração anterior, ou seja, $dp[i-1][w]$.

- Incluir o item: Somamos o valor do item ao valor máximo armazenado para a capacidade restante (**w - peso**), ou seja, $valores[i-1] + dp[i-1][w - peso]$.

Escolhemos a opção que nos dá o maior valor possível.

i \ w	0	1	2	3	4	5
1	0	0	4	4	4	4

Iteração 2: Item com Peso 3 e Valor 5

Neste passo, novamente verificamos se o item 2 com peso 3 pode ser incluído na mochila para cada capacidade **w**. Se couber, temos duas opções:

- Não incluir o item: O valor total na mochila permanece o mesmo que na iteração anterior, ou seja, **$dp[i-1][w]$** .

- Incluir o item: Somamos o valor do item ao valor máximo armazenado para a capacidade restante (**$w - peso$**), ou seja, **$valores[i-1] + dp[i-1][w - peso]$** .

Escolhemos a opção que nos dá o maior valor possível.

$i \setminus w$	0	1	2	3	4	5
2	0	0	4	5	5	9

Iteração 3: Item com Peso 4 e Valor 6

Neste passo, novamente, verificamos se o item 3 com peso 4 pode ser incluído na mochila para cada capacidade **w** . Se couber, temos duas opções:

- Não incluir o item: O valor total na mochila permanece o mesmo que na iteração anterior, ou seja, **$dp[i-1][w]$** .

- Incluir o item: Somamos o valor do item ao valor máximo armazenado para a capacidade restante (**$w - peso$**), ou seja, **$valores[i-1] + dp[i-1][w - peso]$** .

Escolhemos a opção que nos dá o maior valor possível.

$i \setminus w$	0	1	2	3	4	5
3	0	0	4	5	6	9

Resultado Final:

Após o preenchimento completo da matriz **dp** , encontramos o maior valor possível que pode ser carregado na mochila de capacidade 5. Esse valor está localizado em **$dp[3][5]$** .

O maior valor possível que pode ser carregado na mochila é: 9.