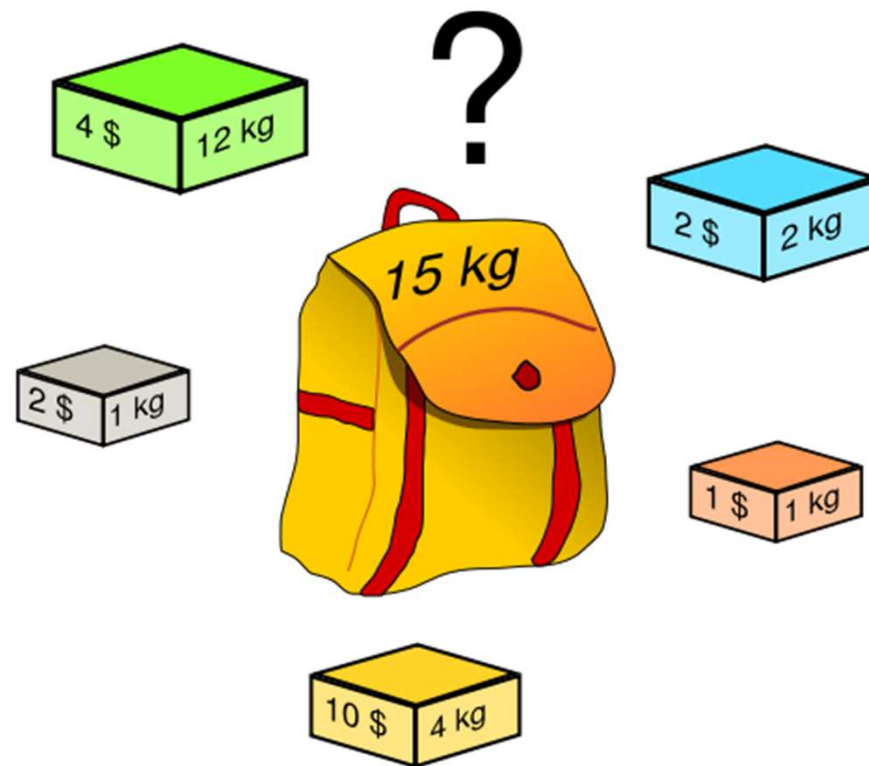


Knapsack[Problema da Mochila]



Knapsack[Problema da Mochila]

- Busca calcular a melhor maneira de se armazenar em um compartimento qualquer muitos itens com valores agregados, de forma que o somatório dos valores de tais itens seja o máximo possível dentre as possibilidades de combinação de itens.
- Usual em supermercados e empresas de armazenagem e logística em geral.

Knapsack[Problema da Mochila]

- Opção 1: Força Bruta
 - Teste todas as combinações possíveis
- Opção 2: Divisão-e-Conquista
 - Divida em problemas menores;
 - Recursivamente resolva estes problemas;
 - Combine os resultados buscando a solução

Knapsack – Como Funciona?

- Vamos criar uma matriz bidimensional chamada `maxTab`, de ordem $N+1 \times C+1$, onde:
- N = número de itens;
- C = Capacidade da mochila;
- A idéia do algoritmo é:
 - começar com uma mochila de capacidade 1 e descobrir o valor máximo possível para esta mochila.
 - Depois se passa para uma mochila de capacidade 2, que aproveitando as informações da mochila de capacidade atual -1 (nesse caso, 1) descobre o valor máximo para uma mochila de capacidade 2.
 - Isso será feito até capacidade atual ser igual à C .
 - Além da capacidade ir crescendo, também vai se adicionando novos itens no processo

Knapsack – Como Funciona?

- **A idéia é:**

percorrer por linha[ou seja, por item] tal tabela, e para a coluna j atual[representa capacidade atual], analisar se o item atual i [linha atual] cabe na mochila de capacidade j . Se couber é preciso escolher o maior valor entre:

- valor na mochila de mesma capacidade j que não tinha esse item[essa informação estará em $\text{maxTab}[i-1][j]$].
- Soma do valor do item com o valor na mochila de capacidade: $(j - \text{peso do item } i)$ e que não tinha esse item($\text{maxTab}[i-1][j - \text{peso do item } i]$)

Então, se couber, teremos:

$\text{maxTab}[i][j] = \text{Máximo}(\text{maxTab}[i-1][j], \text{valor do item} + \text{maxTab}[i-1][j - \text{peso do item } i]);$

Se não couber, significa que o maior valor para essa mochila será o valor da mochila de mesma capacidade, mas que não tem esse item:

$\text{maxTab}[i][j] = \text{maxTab}[i-1][j];$

Knapsack - PseudoCódigo

N = número de produtos;

C = capacidade real da mochila

`itens[N + 1]`; (O índice 0 guarda null)

`maxTab[N+1][C+1]`;

Inicialize com 0 toda a linha 0 e também a coluna 0;

para `i = 1` até N

 para `j = 1` até C

 se item `itens[i].peso` \leq `j` // se o item cabe na mochila atual

`maxTab[i][j] = Max(maxTab[i-1][j],`

`itens[i].valor +`

`maxTab[i-1][j - itens[i].peso])`;

 senão

`maxTab[i][j] = maxTab[i-1][j]`;

retorne `maxTab[N][C]` // valor máximo para uma mochila de capacidade C e
//que pode conter itens que vão do item 1 até o item N.

Exemplo (com / para Teste de Mesa)

- Mochila de 7kg
- 4 peças:

Peso	Valor
5 kg	\$2
2 kg	\$4
2 kg	\$2
1 kg	\$3