**Mochila Fracionária**

Na mochila fracionária, cada item pode ser dividido para aproveitar o espaço.

Texto

Descrição gerada automaticamente

Casos de Teste

1. Entrada:
   1. Itens: {(10, 60), (20, 100), (30, 120)}
   2. Capacidade: 50
   3. Saída Esperada: 240.0
2. Entrada:
   1. Itens: {(5, 50), (10, 60), (20, 140)}
   2. Capacidade: 30
   3. Saída Esperada: 200.0
3. Entrada:
   1. Itens: {(5, 10), (4, 40), (6, 30), (3, 50)}
   2. Capacidade: 10
   3. Saída Esperada: 90.0

**Mochila Fracionária:**

* Complexidade de Tempo: , devido à ordenação inicial dos itens.
* Complexidade de Espaço: , para armazenar os itens.

**Operação Básica**

No Problema da Mochila Fracionária, a operação básica é a multiplicação e soma associada ao cálculo do valor adicionado à mochila para cada item ou fração de item selecionado.

**Expressão de Custo Total**

Somando os custos das etapas descritas:

T(n) = T ordenação + T iteração + T retorno

Substituindo:

T(n) = O (n log n) + c ⋅ n + O (1)

**Forma Detalhada da Expressão**

Considerando os custos constantes (c1, c2, c3) das operações envolvidas, a expressão detalhada do custo seria:

T(n) = c1 ⋅ n log n + c2 ⋅ n + c3

**Assintótica Final**

Para valores grandes de n, o termo dominante é O(n log n). Portanto, a complexidade assintótica do algoritmo é:

T(n) = O (n log n)

Essa expressão reflete que o maior custo está na etapa de ordenação, enquanto o processamento dos itens ocorre linearmente.

**Classe de Problemas**

**1. Classe Geral: Problemas de Otimização**

* O Problema da Mochila Fracionária é um problema de otimização contínua, pois busca maximizar o valor total de itens na mochila, considerando a capacidade da mochila e permitindo que os itens sejam fracionados. Isso significa que podemos pegar uma fração de um item, ao contrário da versão 0/1 da mochila, onde cada item só pode ser incluído ou excluído.

**2. Problema P (Tempo Polinomial)**

* O Problema da Mochila Fracionária pode ser resolvido em tempo polinomial utilizando um algoritmo guloso. Como a solução para esse problema pode ser encontrada eficientemente (em O (n log n) devido à ordenação dos itens, ele pertence à classe P, que representa problemas que podem ser resolvidos em tempo polinomial.

**3. Algoritmo Guloso**

* O Problema da Mochila Fracionária é um exemplo clássico de um problema que pode ser resolvido por um algoritmo guloso. No algoritmo guloso:
* Selecionamos sempre o item com a maior relação valor/peso.
* A solução é construída de maneira a fazer uma escolha ótima localmente em cada passo.
* A característica importante do algoritmo guloso é que ele não retrocede ou faz ajustes em escolhas anteriores.