Conjunto 02 – O Knapsack Problem mochila foi implementado em C++ e aqui seus respectivos casos de teste e análise de custo.

**1. Mochila 0/1 com Programação Dinâmica**

Texto

Descrição gerada automaticamente

**Casos de Teste**

1. Entrada:
   1. Pesos: {2, 3, 4, 5}
   2. Valores: {3, 4, 5, 6}
   3. Capacidade: 5
   4. Saída Esperada: 7
2. Entrada:
   1. Pesos: {1, 2, 3}
   2. Valores: {6, 10, 12}
   3. Capacidade: 5
   4. Saída Esperada: 22
3. Entrada:
   1. Pesos: {1, 3, 4, 5}
   2. Valores: {1, 4, 5, 7}
   3. Capacidade: 7
   4. Saída Esperada: 9

**Mochila 0/1**

* Complexidade de Tempo: onde é o número de itens e é a capacidade da mochila.
* Complexidade de Espaço: (pode ser otimizado para .

**Operação Básica**

No Problema da Mochila 0/1 (0/1 Knapsack), a operação básica é a comparação e atualização de valores na matriz de programação dinâmica dp[i][w], pois é essa etapa que domina o custo computacional do algoritmo.

**Expressão de Custo Total**

Somando as operações das três etapas:

T (n, W) = Inicialização + Preenchimento + Retorno

T (n, W) = (n+1) ⋅ (W+1) + 3⋅n⋅W + O (1)

Expandindo:

T (n, W) = nW + n + W + 1 + 3nW + O (1)

Agrupando os termos dominantes:

T (n, W) = 4nW + n + W + 1

**Análise Assintótica**

Quando n e W são grandes, os termos n e W tornam-se insignificantes comparados a 4nW. Assim, a expressão assintótica é:

T (n, W) = O (n ⋅ W)

**Classe de Problemas**

O Problema da Mochila 0/1 pertence à classe de problemas de otimização combinatória. Mais especificamente, ele faz parte de um conjunto de problemas estudados em teoria da computação e matemática discreta.

**Problemas NP-Completos**

O problema da Mochila 0/1 pertence à classe de problemas NP-completos. Isso significa que:

* É um problema de decisão associado ao problema de otimização: "Existe uma seleção de itens cujo valor total seja pelo menos V, sem exceder a capacidade C?"
* Não se conhece um algoritmo polinomial que resolva o problema no caso geral, a menos que P = NP.