

Programação Dinâmica

INSTRUÇÕES

Para qualquer exercício dessa lista que lhe peça para fornecer um algoritmo/solução para um problema, a menos que explicitamente dito o contrário, você deve:

- descrever em palavras qual é a **ideia** do seu algoritmo. Não é para escrever em prosa o que está escrito no pseudo-código, explique a ideia;
- escrever um pseudocódigo (pode ser em alto nível);
- provar ou fornecer um argumento intuitivo para sua correção. É claro que, se o exercício explicitamente pede uma prova, então um argumento intuitivo não será o suficiente.
 - Observe que em problemas de otimização, duas coisas são importantes:
 - * argumentar que seu algoritmo devolve sempre uma solução viável;
 - * argumentar se o algoritmo é ótimo ou não.
- analisar o tempo do seu algoritmo.

1. No problema da Mochila Inteira são dados n itens, cada item i tendo um valor v_i e um peso w_i , e é dada uma capacidade W da mochila; queremos selecionar um subconjunto $S \subseteq \{1, 2, \dots, n\}$ dos itens cujo peso $\sum_{i \in S} w_i$ seja no máximo W e cujo valor $\sum_{i \in S} v_i$ seja máximo. Perceba que o problema não permite pegar frações dos itens, isto é, só é permitido adicionar à mochila itens por inteiro. Projete um algoritmo de Programação dinâmica Bottom-UP com complexidade $O(nW)$. Você não precisa provar formalmente a correção do seu algoritmo, apenas explique o seu funcionamento, apresente um pseudo-código e analise a sua complexidade.

2. Considere o problema de fazer troco para n centavos usando o menor número total de moedas. Você pode assumir que existe um número infinito de moedas disponíveis para cada valor. Forneça um algoritmo de programação dinâmica que é ótimo para fazer troco tendo disponíveis moedas de valores m_1, m_2, \dots, m_t centavos, para $t \geq 1$, onde algum $m_k = 1$.

3. Execute o algoritmo de programação dinâmica para o problema da Mochila Inteira e mostre a matriz final preenchida pelo mesmo sobre a seguinte entrada: $W = 10, v_1 = 15, w_1 = 3, v_2 = 12, w_2 = 4, v_3 = 17, w_3 = 2, v_4 = 18, w_4 = 5, v_5 = 15, w_5 = 2, v_6 = 10, w_6 = 3, v_7 = 15, w_7 = 1$. Explique como você preencheu a entrada da linha referente ao quarto item e à capacidade 9.

4. Exercício 15.1-3 do livro *Introduction to Algorithms, 3rd Edition* de Cormen, Leiserson, Rivest e Stein.

5. Exercício 15.1-4 do livro *Introduction to Algorithms, 3rd Edition* de Cormen, Leiserson, Rivest e Stein.
6. Dados um grafo G e dois vértices $u, v \in V(G)$, queremos encontrar a distância entre u e v . Mostre que esse problema apresenta subestrutura ótima.
7. Temos n estações ao longo de um rio, numeradas de 1 a n na direção da correnteza. Você pode alugar um barco em qualquer estação i , descer o rio até uma estação $k > i$, devolver o barco e pagar uma taxa $c(i, k)$ pelo passeio. É possível que $c(i, k)$ seja maior que $c(i, j) + c(j, k)$, sendo j uma estação intermediária entre i e k . Nesse caso, é mais barato alugar um barco de i a j e depois outro de j até k . Apresente um algoritmo de programação dinâmica que calcule o custo máximo de uma viagem de 1 a n . Em função de n , quanto tempo o seu algoritmo consome?
8. Leia o enunciado do problema 108 - Maximum Sum e desenvolva um algoritmo para esse problema. Se você quiser, você pode implementar a sua solução e enviá-la para avaliação nesse site, mas não perca muito tempo com isso. Nosso foco é em projeto e análise de algoritmos, não implementação desses. Caso você encontre alguma dificuldade com esse exercício, tente resolver o exercício do rodapé da página¹.

¹Seja $M[0..n, 0..m]$ uma matriz de número inteiros. Dados inteiros i e j , onde $0 \leq i \leq n$ e $0 \leq j \leq m$, desenvolva um algoritmo de programação dinâmica que retorne a soma dos elementos da submatriz $M[0..i, 0..j]$.