

Construção e Análise de Algoritmos

lista de exercícios 21

1. Postos de gasolina 2

Esse problema é, em certo sentido, o contrário do problema que vimos na lista 15.

Quer dizer, ao invés de escolher um subconjunto de postos de gasolina para fazer a parada e abastecer o carro, nós devemos escolher um subconjunto de lugares onde nós vamos construir os postos de gasolina para abastecer os carros.

Mais especificamente, considere uma sequência de localizações (e.g., em quilômetros) ao longo de uma estrada onde os postos podem ser construídos

$$k_1, k_2, k_3, \dots, k_n$$

Para cada localização k_i existe uma demanda d_i pelos serviços do posto de gasolina naquele lugar.

O objetivo, é claro, é atender a maior demanda possível com os novos postos.

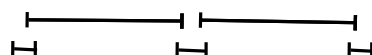
Mas existe a restrição de que não podem ser construídos 2 postos de gasolina a uma distância menor do L quilômetros um do outro.

- a) Apresente um algoritmo de programação dinâmica que encontra uma solução para esse problema.
- b) Estime a complexidade do seu algoritmo.

2. Seleção de atividades 3

Relembre que na lista de exercícios 15, nós consideramos a versão alternativa do problema da seleção de atividades onde o objetivo era maximizar o tempo total em que o laboratório fica ocupado, ao invés de selecionar o maior número de atividades possível.

Abaixo nós temos uma situação onde a mudança no critério de otimização leva a uma solução diferente para o problema



A boa notícia é que agora, com a técnica da programação dinâmica, nós vamos conseguir resolver esse problema.

- a) Apresente um algoritmo que encontra uma solução ótima para esse problema.
- b) Estime a complexidade do seu algoritmo

3. Tirando a graça do jogo

Imagine que n cartas estão organizadas em sequência sobre a mesa

$$c_1, c_2, c_3, \dots, c_n$$

Cada carta c_i tem um certo número de pontos p_i associado a ela, e o objetivo do jogo é obter mais pontos do que o seu adversário.

O jogo se desenrola com os dois jogadores alternando a vez de selecionar uma carta, sendo que apenas a primeira ou a última carta da sequência podem ser selecionadas a cada momento.

A graça do jogo, é claro, está no fato de que em certas situações não é nada fácil saber qual é a melhor escolha a fazer.

Uma estratégia gulosa simples como

- escolher sempre a carta que vale mais pontos

quase certamente levará você a fazer escolhas infelizes.

Mas a técnica de programação dinâmica pode tirar a graça do jogo.

- a) Apresente um algoritmo que computa uma tabela J que indica a jogada ótima a ser feita para cada situação possível do jogo, a partir da sequência de cartas c_1, \dots, c_n .

A ideia aqui é que $J[i, j]$ indica a jogada ótima na situação em que as cartas nos dois extremos da sequência são c_i e c_j — por exemplo, com o 0 indicando a seleção de c_i e o 1 indicando a seleção de c_j .

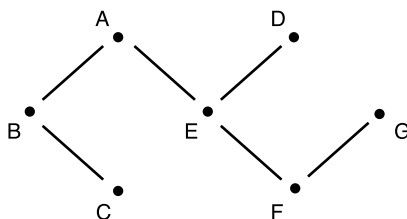
- b) Estime a complexidade do seu algoritmo.

4. Cobertura de vértices em árvores

O problema da cobertura de vértices é mesmo um problema bem difícil.

Mas, em alguns casos, como a situação onde o grafo é uma árvore (i.e., não contém ciclos), ele pode ser resolvido de maneira ótima.

Abaixo nós temos um exemplo simples com solução ótima de tamanho 3: $\{B, E, G\}$.



Apresente um algoritmo de programação dinâmica que encontra uma cobertura de vértices de tamanho mínimo em uma árvore G .

5. Tabuleiros e feijões

Considere um tabuleiro com 4 linhas e n colunas, onde cada casa do tabuleiro está associada a um número inteiro.










Abaixo nós temos um exemplo simples, com $n = 6$

1	7	-8	-2	3	9
8	9	11	1	-5	4
4	-2	12	2	-7	2
-5	0	-3	-1	9	10

O problema consiste em marcar um subconjunto de posições do tabuleiro (digamos, com feijões), de modo que

- não existem duas posições marcadas vizinhas uma da outra (i.e., lado a lado, ou uma em cima da outra)
- a soma dos números associados às posições marcadas é a maior possível

Abaixo nós temos uma solução para o tabuleiro exemplo (não necessariamente ótima)

	7	-8	-2	3	
8		11		-5	4
	-2		2	-7	
-5		-3	-1		10

Note que você não precisa marcar duas posições em todas as colunas, e pode mesmo deixar algumas colunas vazias se quiser.

- Apresente um algoritmo de programação dinâmica que encontra uma solução ótima para esse problema.
- Estime a complexidade do seu algoritmo.