

Construção e Análise de Algoritmos

lista de exercícios 23

1. Programando a viagem

Imagine que você vai fazer uma longa viagem.

Você vai começar no quilômetro k_0 da estrada, e está indo passar uma temporada em um hotel que fica no quilômetro k_n .

Como o hotel é bem distante, você terá que fazer diversas paradas no meio do caminho. Examinando um mapa, você seleciona uma coleção de lugares onde poderia passar a noite

$$k_1, k_2, k_3, \dots, k_{n-1}$$

Mas, como todo estudante de computação, você é uma pessoa sistemática e enfiou na cabeça que seria ideal viajar aproximadamente L quilômetros por dia, mas nunca menos do que isso!

Agora você tem o problema de escrever um programa de computador que escolhe locais k_{j_1}, \dots, k_{j_h} para fazer as paradas, de modo que a seguinte expressão seja minimizada:

$$\sum_{i=1}^{h+1} [(k_{j_i} - k_{j_{i-1}}) - L]^2$$

satisfazendo a restrição

$$k_{j_i} - k_{j_{i-1}} \geq L, \text{ para } i = 1, \dots, h+1$$

onde nós estamos assumindo que $k_{j_0} = k_0$ e $k_{j_{h+1}} = k_n$.

2. Mínimos quadrados segmentado 2

Em alguns casos, informações específicas do contexto podem ser suficientes para determinar que a nossa coleção de pontos $P = \{p_1, p_2, \dots, p_n\}$ foi gerada por K processos diferentes (onde os pontos gerados por processos diferentes se encontram em faixas relativamente separadas umas das outras).

Nesse caso, é natural querer construir uma aproximação para os pontos formada por exatamente K linhas — mesmo que o erro possa ser reduzido usando mais linhas.

- Apresente um algoritmo de programação dinâmica que encontra uma aproximação com K linhas para uma coleção de pontos P , com o menor erro possível.
- Estime a complexidade do seu algoritmo.

3. A biblioteca

A biblioteca tem uma coleção de livros L_1, L_2, \dots, L_n que precisam ser colocados em estantes. Cada livro L_j tem uma altura a_j e uma largura l_j . E as estantes possuem todas uma largura fixa K , mas a distância entre as prateleiras pode ser ajustada para acomodar livros com alturas diferentes.

Depois de alguma experimentação, os bibliotecários perceberam que, ao invés de colocar o máximo de livros em cada prateleira, poderia valer a pena mover livros para outra prateleira para reduzir a utilização de espaço vertical.

Mas, é claro que isso só pode ser feito sem tirar os livros da ordem.

Assim, o problema foi definido da seguinte maneira.

- Particionar a sequência de livros L_1, \dots, L_n em prateleiras P_1, \dots, P_m de modo que
 - a soma das larguras em cada prateleira P_i não excede K
 - a soma das alturas de todas as prateleiras é a menor possível

onde a altura de uma prateleira é dada pelo tamanho do livro mais alto que foi colocado nela

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

4. Reconstrução de texto

Imagine que você tem uma sequência de caracteres

$$X = x_1 \ x_2 \ x_3 \ \dots \ x_{n-1} \ x_n$$

que você acredita ser um texto, onde todos os espaços em branco e sinais de pontuação foram perdidos — por exemplo, “Eraumaveschapauzinvermeieobobomau...”.

Você deseja reconstruir o documento utilizando um dicionário eletrônico, que responde consultas no seguinte formato

$$\text{dic}(p) = \begin{cases} 1 & , \text{ se } p \text{ é uma palavra do dicionário} \\ 0 & , \text{ caso contrário} \end{cases}$$

O dicionário não é perfeito, e você também sabe que o texto contém erros ortográficos.

Apesar disso, você acredita que uma separação dos caracteres que maximiza a proporção de palavras que estão no dicionário pode dar uma boa reconstrução do texto.

Mais precisamente, para uma segmentação qualquer do texto

$$S = p_1, p_2, p_3, \dots, p_k$$

nós definimos a sua qualidade como

$$Q(S) = \frac{1}{|S|} \cdot \sum_j \text{dic}(p_j)$$

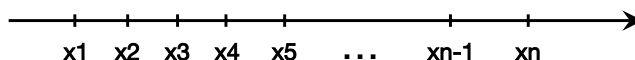
onde $|S|$ é o número de palavras em S .

O objetivo do problema é encontrar uma segmentação S com o maior valor $Q(S)$ possível.

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

5. k -cobertura de intervalos unitários

Lembre que uma *cobertura de intervalos unitários* para uma sequência de pontos



consiste em uma coleção de intervalos $[a_1, b_1], [a_2, b_2], \dots, [a_m, b_m]$ tal que

- i. $b_j - a_j = 1$, para todo j (i.e., todos os intervalos tem tamanho 1)
- ii. todo ponto x_i pertence a algum intervalo $[a_j, b_j]$

Agora, imagine que você só pode usar K intervalos, e que isso não é suficiente para cobrir todos os pontos.

Além disso, suponha que cada ponto x_i está associado a um valor v_i , que pode ser positivo ou negativo.

O valor de uma cobertura de k intervalos unitários é dado pela soma dos valores dos pontos que estão cobertos pelos intervalos.

E o objetivo do problema é encontrar uma cobertura de k intervalos unitários com o maior valor possível.

(Note que, apesar de você ter K intervalos, você não precisa usar todos eles para cobrir os pontos. Por exemplo, se todos os pontos possuem valor negativo, a melhor coisa a fazer é colocar os intervalos fora da faixa dos pontos, deixando todos eles descobertos.)

- a) Apresente um algoritmo de programação dinâmica para esse problema.
- b) Estime a complexidade do seu algoritmo. o problema.