Universidade Federal do Ceará
Centro de Ciências
Mestrado e Doutorado em Ciência da Computação

Construção e Análise de Algoritmos

Lista de exercícios 2

1. Altere os algoritmos de entrelaçamento e de ordenação por entrelaçamento (sem alterar as complexidades!) para obter um algoritmo que resolva o seguinte problema: dado um vetor sequencial contendo n números inteiros positivos e um outro número inteiro positivo x, determine se existem ou não dois elementos em S cuja soma seja exatamente x.

2. Descreva um algoritmo para resolver o problema de determinar o segundo menor elemento de um vetor S. Determine exatamente o número de comparações efetuadas pelo algoritmo. (Ele será considerado tão melhor quanto menor for este número.)

3. Suponhamos que em vez de selecionar o k-ésimo menor elemento de um vetor com n elementos, estamos interessados em determinar os k menores elementos (sem nos preocuparmos com a ordem relativa entre eles). Será que isto pode ser feito em tempo $O(n)$?

4. Elabore um algoritmo de decomposição de um vetor S em três subvetores. Esse algoritmo recebe como entrada, além do vetor S, um valor piv pertencente a S, e os índices p e r, $1 \leq p \leq r$. O algoritmo deve rearrumar os elementos em $S[p \dots r]$ e retornar dois índices i e j satisfazendo as seguintes propriedades:

   (a) se $p \leq k \leq i$, então $S[k] < piv$;

   (b) se $i < k \leq j$, então $S[k] = piv$;

   (c) se $j < k \leq r$, então $S[k] > piv$.

   A complexidade do seu algoritmo deve ser $O(n)$.

5. Reescreva o algoritmo de seleção de complexidade $O(n)$ para que ele aceite elementos repetidos.

6. Elabore um algoritmo que, dado um vetor S com $n > 0$ elementos, retorne um vetor V, de tamanho n, com a seguinte propriedade: $V[i]$ é o número de ocorrências de $S[i]$ em S. A complexidade do seu algoritmo deve ser $O(n \log n)$.

7. Dados dois vetores binários B_1 e B_2, ambos de tamanho $n > 0$, a distância de Hamming entre B_1 e B_2 é definida como sendo o número de índices $1 \leq i \leq n$ tais que $B_1[i] \neq B_2[i]$. Um Código de Gray de ordem n é uma sequência $B^n = \langle [B^n_0], \dots, [B^n_{2^n-1}] \rangle$ dos 2^n vetores binários distintos de tamanho n tal que a distância de Hamming entre dois vetores consecutivos quaisquer na sequência é 1. Por exemplo, $\langle [1,1], [1,0], [0,0], [0,1] \rangle$ é um Código de Gray de ordem 2.

   (a) Provar, por indução, que

   $$\begin{aligned} G^1 &= \langle [0], [1] \rangle \\ G^n &= \langle [0, G^{n-1}_0], \dots, [0, G^{n-1}_{2^{n-1}-1}], [1, G^{n-1}_{2^{n-1}-1}], \dots, [1, G^{n-1}_0] \rangle, n > 1 \end{aligned}$$

   é um Código de Gray de ordem n.

   (b) Elaborar um algoritmo recursivo que determine um Código de Gray de ordem n em $O(2^{n-m})$ passos a partir de um Código de Gray de ordem m, $1 \leq m < n$.