

MC448 – Análise de Algoritmos Lista de Exercícios 4

Orlando Lee

O livro do Manber (capítulos 5 e 6) possui vários exemplos e exercícios sobre projeto de algoritmos por indução. Alguns exercícios estão resolvidos no final do livro.

Leia: Procure desenvolver os algoritmos com a complexidade pedida ou com a menor complexidade possível. Tente usar o método de projeto de algoritmos por indução sempre que possível. Deixe clara qual foi a **hipótese de indução usada**. Você **deve justificar** porque seu algoritmo tem a complexidade pedida ou descrita por você.

1. Suponha que seja dada uma árvore A que não é AVL. Um nó é chamado **nó AVL** se seu fator de balanceamento é -1 , 0 ou 1 . Projete um algoritmo que identifique todos os nós em A que **não** são nós AVL, mas todos seus descendentes são nós AVL. Qual a complexidade de seu algoritmo?
2. Seja $A[1..n]$ um vetor de números reais e seja x um número real. Projete um algoritmo de complexidade $O(n \lg n)$ que devolve (se existir) dois elementos distintos $A[i]$ e $A[j]$ tais que $A[i] + A[j] = x$.
3. Seja a_1, \dots, a_n uma seqüência de números reais. Projete um algoritmo de complexidade de tempo $\Theta(n)$ que determina uma subseqüência consecutiva cujo produto de seus elementos seja máximo. O produto dos elementos da subseqüência vazia é 1 por definição. Por exemplo: para a seqüência $-1, 1/2, 4, -1/3, 1/2, -6, 1/4, 5, 1/4$ a subseqüência de produto máximo é $4, -1/3, 1/2, -6, 1/4, 5$.
4. Seja M uma matriz $n \times m$ de números reais tal que
 - (a) cada linha de M está ordenada em ordem crescente (da esquerda para a direita) e
 - (b) cada coluna de M está ordenada em ordem crescente (de cima para baixo).Descreva um algoritmo (baseado em comparações) que recebe M e um inteiro x e determina se x aparece em M (ou seja, se existem índices i, j tais que $M[i, j] = x$). Seu algoritmo deve ter complexidade $O(n + m)$.
5. Considere um vetor $A[1..n]$ com a seguinte propriedade: para todo i , $1 \leq i \leq n - 1$, vale que $|A[i] - A[i + 1]| \leq 1$. Suponha que seja dado um elemento x tal que $A[1] \leq x \leq A[n]$. Projete um algoritmo eficiente para encontrar um índice j tal que $A[j] = x$ (sempre existe tal j ??). Qual é a complexidade do seu algoritmo?
6. A entrada consiste de k vetores ordenados de números reais. Projete um algoritmo para intercalar todos esses vetores e obter um vetor ordenado. Seu algoritmo deve ter complexidade $O(n \lg k)$ onde n é o número total de elementos em todos os vetores. Note que os vetores não precisam ter o mesmo tamanho.
7. Considere o seguinte problema: dados n intervalos na reta real, $I_i = [a_i, b_i]$, $i = 1, 2, \dots, n$, projete um algoritmo que lista todos os intervalos que estão contidos dentro de pelo menos um dos outros intervalos passados na entrada. O seu algoritmo deve ter complexidade $O(n \lg n)$.
8. Diz-se que um ponto $p = (x_p, y_p)$ do plano **domina** um outro ponto distinto $q = (x_q, y_q)$ do plano se $x_p \geq x_q$ e $y_p \geq y_q$. Um ponto p é **maximal** em relação a um conjunto de pontos S se $p \in S$ e nenhum ponto de S domina p .

Projete um algoritmo de complexidade $O(n \lg n)$ para encontrar todos os pontos maximais de um dado conjunto S de n pontos no plano.