

Algoritmos e Complexidade

Exame Especial — 2. Parte — 12 de Setembro de 2008

1 – Considere um *array* ordenado v , de n elementos inteiros ($n \geq 1$). Os elementos encontram-se representados em ordem **crescente**.

Pretende-se verificar, usando a estratégia de **pesquisa binária**, se um dado elemento x pertence ao *array* v e determinar o seu índice, caso lhe pertença.

a) Construa uma função recursiva que determine o índice do elemento procurado, caso exista, usando pesquisa binária.

b) Determine o número de *comparações* — associadas a elementos do *array* — efectuadas pelo algoritmo da função anterior no **Pior Caso**.

Em que situações ocorre o Pior Caso?

c) Seja $n = 2^k - 1$, $k = 1, 2, 3, \dots$. Faça, agora, a análise do número de *comparações* — associadas a elementos do *array* — efectuadas pelo algoritmo da função anterior no **Caso Médio**.

Atenção: Nesta análise, deve assumir que o elemento procurado se encontra no *array*.

Relações úteis:

$$\sum_{i=0}^{k-1} 2^i = 2^k - 1 \qquad \sum_{i=0}^{k-1} i 2^i = 2 + 2^k (k - 2)$$

2 – Considere o tipo abstracto de dados **Árvore Binária de Inteiros**, em cujos nós é possível armazenar um número inteiro.

Considere também que os números inteiros se encontram registados “**em-ordem**” **crescente**.

Elabore funções eficientes que permitam:

a) Encontrar, de modo **iterativo**, o **maior** elemento armazenado numa árvore dada.

b) Determinar, de modo **recursivo**, o número de nós com **dois descendentes directos** — i.e., com dois filhos — de uma árvore dada.

c) Inserir — **ordenadamente** e de modo **recursivo** — um dado número inteiro numa árvore, caso não esteja já registado na árvore.

3 – Considere o tipo abstracto de dados **Digrafo**, definido usando uma estrutura de dados dinâmica que representa um dado digrafo $G(V, E)$, com n vértices e m arestas, armazenando a **lista dos vértices** do digrafo e associando a cada elemento dessa lista (i.e., a cada vértice) a correspondente **lista de adjacências**.

Considere também que os n vértices de um digrafo se encontram identificados pela sequência de números inteiros $0, 1, \dots, (n-1)$, e que há um valor real não-negativo (i.e., uma “*distância*”) associado a cada aresta de um digrafo.

Dado um vértice $v_i \in V$, pretende-se determinar a **Árvore dos Caminhos Mais Curtos** com raiz em v_i .

Essa árvore representa os caminhos mais curtos, definidos no digrafo G , entre v_i e cada um dos outros vértices do digrafo que são alcançáveis a partir de v_i .

a) Desenvolva uma função que, usando uma estratégia **iterativa**, permita determinar a *Árvore dos Caminhos Mais Curtos*.

b) Desenvolva agora uma função que, usando uma estratégia **recursiva**, permita determinar a *Árvore dos Caminhos Mais Curtos*.

Atenção:

- Não se esqueça de que o digrafo pode conter ciclos.
- Assuma que estão definidos os tipos abstractos **Pilha** e **Fila**; não é necessário implementá-los, caso os queira utilizar.
- Desenvolva eventuais funções auxiliares de que possa necessitar.