

Algoritmos e Complexidade

Exame de Recurso — 2. Parte — 7 de Julho de 2008

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

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 iterativa 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:

$$\sum_{i=0}^{k-1} 2^i = 2^k - 1$$

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

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) Determinar, de modo **recursivo**, o número de **nós intermédios** — i.e., tendo pelo menos um descendente — de uma árvore dada.

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

c) Listar, de modo **recursivo**, os n primeiros elementos armazenados numa dada 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 $0, 1, \dots, (n - 1)$.

Dado um vértice $v_i \in V$, pretende-se listar todos os vértices v_j alcançáveis a partir de v_i .

Essa listagem deverá ser iniciada pelo vértice v_i e ser ordenada — de modo não-decrescente —, de acordo com a distância associada ao caminho mais curto de v_i para cada v_j (i.e., o número de arcos que o definem).

Por exemplo, para um dado digrafo e para o caso dos vértices alcançáveis a partir do vértice 1, obter-se-ia:

No	1	-	Distancia	0
No	2	-	Distancia	1
No	3	-	Distancia	1
No	4	-	Distancia	2
No	5	-	Distancia	2
No	6	-	Distancia	3

Desenvolva uma função que, dado um digrafo e um seu vértice v_i , lhe permita obter a listagem desejada.

Atenção:

- O digrafo pode conter ciclos.
- Assuma que estão definidos os tipos abstractos **Pilha** e **Fila**; não é necessário implementá-los.
- Desenvolva eventuais funções auxiliares de que possa necessitar.