

Algoritmos e Complexidade

Exame Normal — 2. Parte — 20 de Junho de 2008

1 – O Algoritmo “**Bubble Sort**” é um dos métodos possíveis para efectuar a ordenação de um vector e consiste, basicamente, em comparar, repetida e sucessivamente, elementos adjacentes de um (sub-)vector e realizar a sua “troca”, se necessário.

Pretende-se ordenar um vector de números inteiros, com possíveis elementos repetidos, de modo **não-crescente**.

a) Construa uma função que percorra uma vez os elementos de um vector $v[0..n-1]$, $n > 1$, comparando e ordenando elementos adjacentes. A função deverá devolver um inteiro, 1 ou 0, indicando se foi ou não necessário “trocar” algum par de elementos adjacentes.

b) Faça a análise do número de **comparações** e de **trocass** — entre elementos do vector — efectuadas pelo algoritmo da alínea anterior, no **melhor caso** e no **pior caso**.

c) Construa agora outra função que, usando a função anterior, implemente o Algoritmo “**Bubble Sort**”.

d) Faça a análise do número de **comparações** e de **trocass** — entre elementos do vector — efectuadas pelo algoritmo de ordenação da alínea anterior, no **melhor caso** e no **pior caso**.

2 – O problema das **Torres Duplas de Hanói** consiste numa variação simples do problema original: a torre inicial é formada neste caso por $2n$ discos com n diâmetros diferentes, havendo um par de discos de cada tamanho.

Como habitualmente, os discos que compõem a torre inicial estão empilhados ordenadamente pelos seus diâmetros (os mais pequenos no topo) num de três postes, pretendendo-se transferir a torre para um dos outros dois postes, mas deslocando apenas um só disco de cada vez e nunca colocando um disco de maior diâmetro sobre outro mais pequeno.

Considere que quaisquer dois discos de igual diâmetro são indistinguíveis um do outro.

a) Construa uma função que permita estabelecer os movimentos necessários para transferir uma torre com $k = 2n$, $n = 1, 2, \dots$, discos entre quaisquer dois dos três postes.

b) Determine uma expressão para o número total de movimentos de discos que é necessário efectuar. Qual é a ordem de complexidade do algoritmo da alínea anterior?

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

Além dos ponteiros para as suas sub-árvores, cada nó contém um ponteiro adicional para o seu nó progenitor.

Desenvolva funções eficientes que permitam:

a) Dado um ponteiro para um nó de uma árvore, determinar o seu nível. ✕ ✓

b) Dado um ponteiro para um nó de uma árvore, obter o ponteiro para o nó que se lhe segue na travessia em “pré-ordem”.