

Programação Imperativa

LEI + LCC (1º ano)

6ª Ficha Prática

Ano Lectivo de 2006/07

Objectivos

O objectivo desta ficha é treinar o aluno em algoritmos de ordenação de arrays.

Exercícios

1. Desenvolva um algoritmo, e codifique-o em linguagem C, para ordenar um array recorrendo ao seguinte método: percorrer o array diversas vezes, colocando no início do array, em cada iteração, o menor elemento da lista.

Nota: Este algoritmo é chamado de trocas directas.

Desenvolva depois a variante *bubble-sort* do anterior, em que se pára logo que não haja mais trocas.

2. Desenvolva um algoritmo, e codifique-o em linguagem C, para ordenar um array recorrendo ao seguinte método:
 - Dividir o array não ordenado em dois sub-arrays de aproximadamente metade do tamanho;
 - Ordenar cada um dos dois sub-arrays;
 - Juntar os dois sub-arrays ordenados em um único array ordenado.

Nota: Este algoritmo é chamado de *merge-sort*, também conhecido como um algoritmo de *divide and conquer* — dividir o problema inicial em dois sub-problemas e resolver um problema menor utilizando recursividade.

3. Desenvolva um algoritmo, e codifique-o em linguagem C, para ordenar um array recorrendo ao seguinte método:
 - O array $A[p..r]$ é subdividido em dois arrays $A[p..q]$ e $A[q+1..r]$ não vazios tal que cada elemento de $A[p..q]$ é menor ou igual a cada elemento de $A[q+1..r]$. O índice q é calculado como parte desta divisão.
 - Os dois sub-arrays $A[p..q]$ e $A[q+1..r]$ são ordenados por recursividade.

Nota: Este algoritmo é chamado de *quick-sort* e é também um algoritmo de *divide and conquer*.

4. Desenvolva um algoritmo, e codifique-o em linguagem C, para fazer a inserção ordenada de elementos numéricos num array.