

Estruturas de Dados

Lista 3 - Algoritmos de Ordenação

1. Implemente cada um dos métodos de ordenação a seguir e informe o número de trocas (em função de n) que cada um deles faz:
 - (a) Selection Sort
 - (b) Bubble Sort
 - (c) Insertion Sort
2. Dados dois vetores ordenados, contendo cada um deles n números inteiros, deseja-se uní-los em um outro vetor maior, contendo $2n$ números, que também serão armazenados de forma ordenada. Argumente qual será a complexidade de tempo desse processo na estratégia mais eficiente.
3. Apresente o pseudocódigo de uma função que ordena uma lista encadeada. Qual a complexidade do seu algoritmo?
4. Apresente o pseudocódigo de uma função `RANDOMIZEDPARTITION(V, n)`. Essa função é similar à função `PARTITION` apresentada em sala, porém ela seleciona um elemento aleatório do vetor como pivô e faz o particionamento utilizando esse valor. Para isso, considere uma função `RANDOM(n)` que retorna um valor aleatório entre 1 e n . Se desejar, você também pode utilizar a função `PARTITION(V, n)` vista em sala, que seleciona o primeiro elemento do vetor ($V[1]$) como pivô para fazer o particionamento.