



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Estruturas de Dados e Algoritmos
Lista de Exercícios – Ordenação
Prof. Daniel Saad Nogueira Nunes

Aluno: _____
Matrícula: _____

Exercício 1

Implemente os seguintes métodos de ordenação para vetores de inteiros:

- (a) Bubblesort;
- (b) Insertionsort;
- (c) Mergesort;
- (d) Quicksort;
- (e) Heapsort;
- (f) Countingsort;
- (g) Radixsort.

Exercício 2

Faça o mesmo do exercício anterior (exceto para Countingsort e Radixsort), mas para vetores de *strings*.

Exercício 3

Compare os métodos de ordenação com relação ao tempo gasto para ordenar um vetor de inteiros. Certifique-se de utilizar funções para aferir o tempo gasto pelo algoritmo.

Exercício 4

Compare as implementações do quicksort utilizando pivôs fixos e pivôs aleatórios. Utilize funções para cronometrar o tempo de execução do seu programa.

Exercício 5

Elabore um algoritmo que, dado um vetor de tamanho n e um parâmetro $k < n$, responda quais são os k menores elementos do vetor da maneira mais eficiente que você encontrar.

Dica: utilize o conceito de heap.

Exercício 6

A ordenação por classificação sobre vetores inteiros contendo valores na faixa $[a, b]$ funciona da seguinte maneira:

-
- Crie um vetor c de tamanho $[b - a + 1]$.
 - Defina $c[i - a]$ como o número de vezes que o número i aparece no vetor v .
 - Utilize c para ordenar o vetor v ao inspecioná-lo da esquerda para direita e inserir elementos em v enquanto $c[i]$ é positivo. A cada elemento i inserido, $c[i]$ é decrementado.

Implemente um algoritmo utilizando a descrição acima.

Exercício 7

O método de ordenação por seleção quadrática funciona da seguinte maneira:

- 1) Divida o vetor logicamente em grupos de \sqrt{n} elementos.
- 2) Encontre o maior elemento de cada grupo e insira-o em um vetor auxiliar.
- 3) Encontre o maior elemento nesse vetor auxiliar e o coloque no vetor solução.
- 4) Descarte este maior elemento no grupo em que ele se encontrava.
- 5) Repita os passos 2 a 5 até que o vetor solução tenha sido totalmente preenchido.

Implemente a ordenação por seleção quadrática da maneira mais eficiente possível.

Exercício 8

Projete um algoritmo que faça o merge de k vetores ordenados de tamanho n da maneira mais eficiente possível.

Dica: utilize a estrutura de dados Heap.

Exercício 9

Implemente a função `quick_find`, que utiliza o PARTITION do QUICKSORT para encontrar o k -ésimo menor elemento de um vetor v .

Exercício 10

Implemente a busca binária para vetores de inteiros;

Exercício 11

Implemente a busca binária para vetores de *strings*.

Exercício 12

Compare o desempenho da busca binária e da busca sequencial. Certifique-se de utilizar funções para monitorar o tempo gasto por cada um dos procedimentos.

Exercício 13

Implemente a ordenação e busca binária utilizando as funções `qsort` e `bsearch` da biblioteca padrão.