



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

Aluno: \_\_\_\_\_

Matrícula: \_\_\_\_\_

### Exercício 1

Dê em pseudocódigo os algoritmos de ordenação abaixo:

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

### Exercício 2

Projete um algoritmo que faça o merge de  $k$  vetores ordenados de tamanho  $n$  cada em tempo  $O(nk \lg k)$ .

**Dica: utilize o conceito de heap. Assuma que você tem prontas todas as funções que atuam sobre a Heap.**

### Exercício 3

O professor Inácio Firmino Bueno afirmou que conseguiu realizar o merge de dois vetores em tempo  $\Theta(n^{0.99})$  usando um algoritmo baseado em comparações. Isso é possível?

### Exercício 4

O professor Inácio garante que é possível ordenar em tempo  $o(n \lg n)$  utilizando comparações através do Heapsort com Heaps de Fibonacci. Mostre que ele está errado.

### Exercício 5

O prof. Inácio afirmou que o seu novo método de ordenação consegue ordenar qualquer vetor de 5 elementos com apenas 6 comparações. O prof. Inácio está falando a verdade ou ele é um picareta?

---

## Exercício 6

Projete um algoritmo capaz de encontrar todos os elementos que se repetem mais que  $\frac{n}{3}$  vezes. Este algoritmo deve ter custo  $O(n)$ .

## Exercício 7

Projete um algoritmo capaz de encontrar todos os elementos que se repetem mais que  $\frac{n}{k}$  vezes para um parâmetro  $k$ . Este algoritmo deve ter custo  $O(nk)$ .

## Exercício 8

implemente um algoritmo que resolva o problema anterior com custo  $O(n \lg k)$ .

## Exercício 9

Elabore um algoritmo que, dado um vetor de tamanho  $n$  e um parâmetro  $k < n$ , responda quais são os  $k$  menores elementos do vetor em tempo  $O(n \lg k)$ .

## Exercício 10

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.

Projete um algoritmo utilizando a descrição acima e analise a sua complexidade.

## Exercício 11

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.

Escreva e analise um algoritmo que implemente a ordenação por seleção quadrática em  $O(n^{\frac{3}{2}})$ .

## Exercício 12

É possível implementar o método de ordenação por seleção quadrática em tempo  $O(n \lg n)$ ? Se sim, descreva como.

## Exercício 13

Implemente o algoritmo QUICK-FIND que utiliza o PARTITION do QUICKSORT para encontrar o  $k$ -ésimo menor elemento de um vetor  $V$ .

---

## Exercício 14

Mostre como o Quicksort pode ser feito em tempo de pior caso  $O(n \log n)$ .

**Dica:** escolha bem o pivô.