UFRN-IMD
BTI
IMD0030 – LP1

Aula08 - Exercícios: Algoritmos de ordenação (parte 1)

1. Escreva uma função com a seguinte interface:

```
int soma( int (*f)(int), int inicio, int fim );
```

Uma chamada **soma(g, i, j)** deve retornar **g(i) + g(i + 1) + ... + g(j)**. Faça diversas dessas chamadas em uma função **main()** e imprima o resultado de cada uma delas.

2. Implemente uma função que permita a ordenação crescente/decrescente através do **Selection Sort** explorando o conceito de ponteiros de funções.

3. Implemente a versão iterativa e a versão recursiva do **Bubble Sort**. Faça a ordenação do vetor A = {12, 23, 2, 1, 5, 67, 10, 11, 20, 22} utilizando as duas versões e mostre qual abordagem é mais vantajosa empiricamente (número de comparações).

4. Suponha que temos uma sequência S de n elementos, cada um deles pintado de azul ou de vermelho. Suponha que S é representado por um arranjo e forneça uma função capaz de ordenar S de forma que todos os elementos azuis sejam listados antes de todos os elementos vermelhos.

5. Você pode estender sua abordagem para a questão anterior de forma a tratar com três cores?

6. (Desafio) Suponha que temos uma sequência S de n elementos onde existe uma relação de ordem total. Descreva um método eficiente para determinar se existem dois elementos iguais em S. Qual o tempo de execução de seu algoritmo? Demonstre sua resposta empiricamente.

7. (Desafio) Suponha que temos uma sequência S de n elementos, de forma que cada elemento em S representa um voto diferente para líder estudantil, dado como um inteiro representando o número de matrícula do candidato escolhido. Sem assumir hipóteses sobre os candidatos ou o número de candidatos, implemente:

a) uma função para determinar quem vence a votação representada por S supondo que o candidato com o maior número de votos será o escolhido.

b) uma função para imprimir a lista completa dos candidatos, por ordem de número de votos recebidos.