

**Disciplina:** Estrutura de dados  
**Professor:** Atílio Gomes  
**Curso:** Engenharia de Computação  
**Aluno:** Marcos Paulo Rocha Rodrigues  
**Matrícula:** 403581

## Algoritmos de ordenação

### Resumo

Este relatório tem como objetivo descrever como foi a experiência na implementação dos seguintes algoritmos de ordenação, Insertion Sort, Selection Sort, Merge Sort e Quick Sort para a disciplina de Estrutura de dados, usando a linguagem C++ e implementado como TAD(Tipos Abstratos de Dados). O projeto foi desenvolvido individualmente.

## 1 Insertion Sort

### Descrição do algoritmo

O algoritmo compara dois a dois cada elemento do vetor comparando quem dos dois elementos é o menor e colocando o menor sempre a esquerda. Assim insere o elemento do vetor atual( $v[i]$ ) na posição correta.

Figura 1: Insertion Sort Iterativo

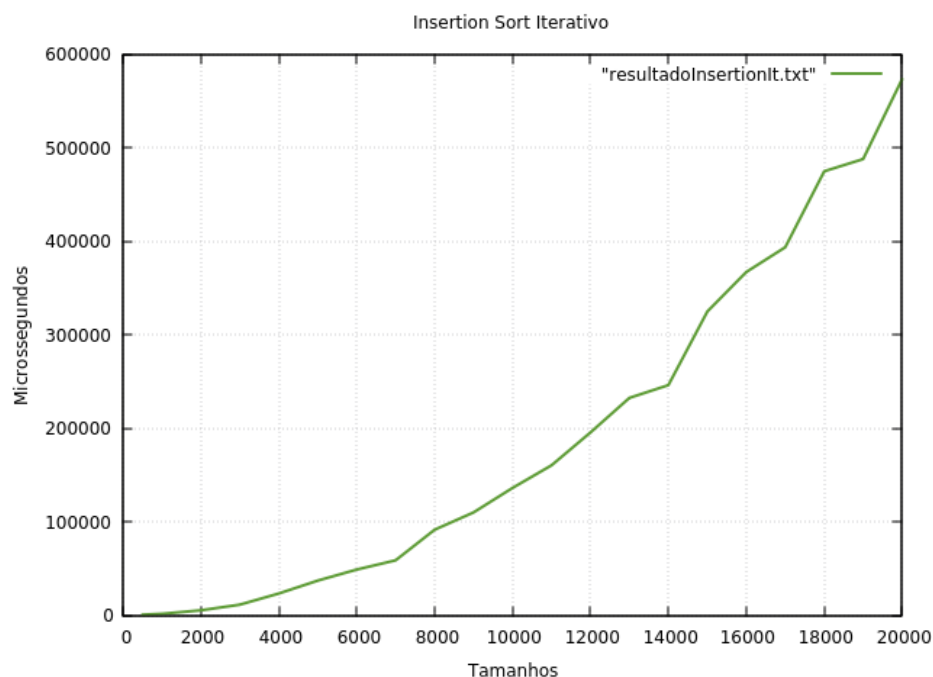


Figura 2: Insertion Sort Recursivo

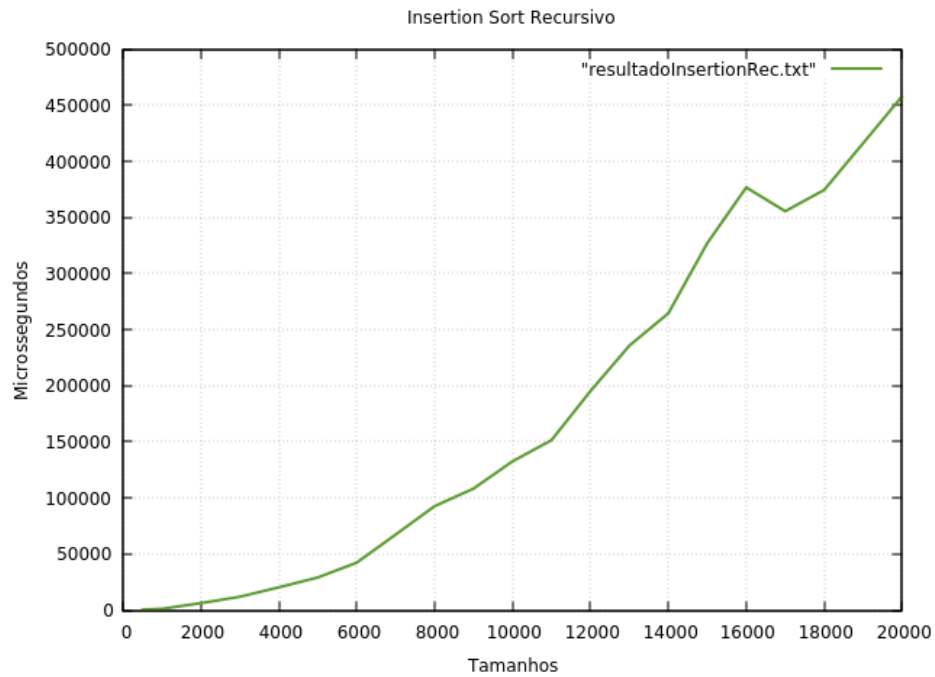
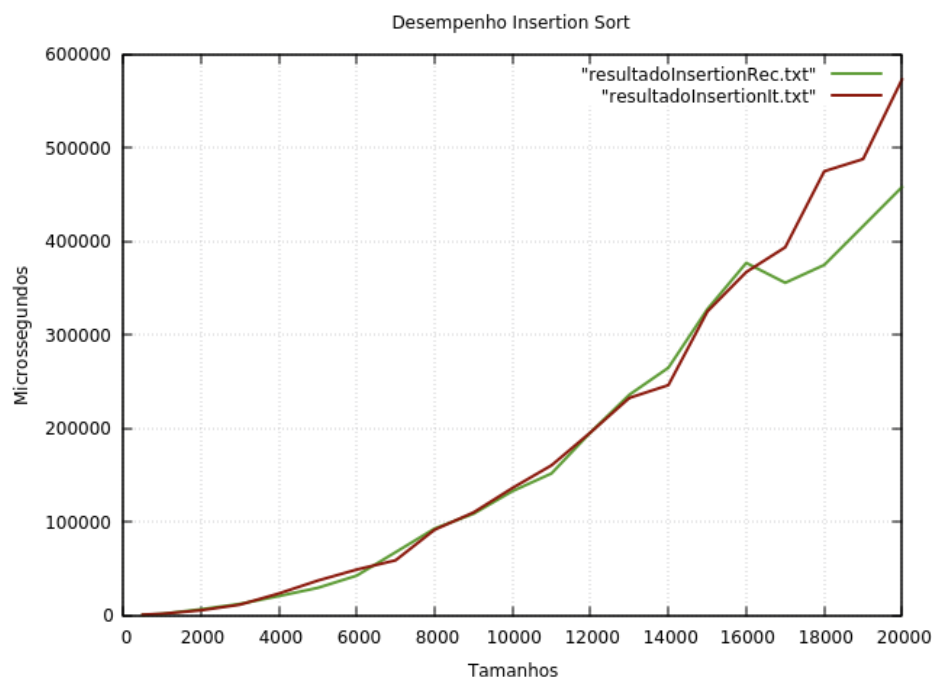


Figura 3: Comparação de desempenho



## 2 Selection Sort

### Descrição do algoritmo

O algoritmo, resumidamente, seleciona o menor valor do vetor e coloca em ordem crescente, por exemplo, o primeiro menor valor encontrado vai ficar na primeira posição do vetor, o segundo menor valor encontrado vai ficar na segunda posição do vetor e assim por diante.

Figura 4: Selection Sort Iterativo

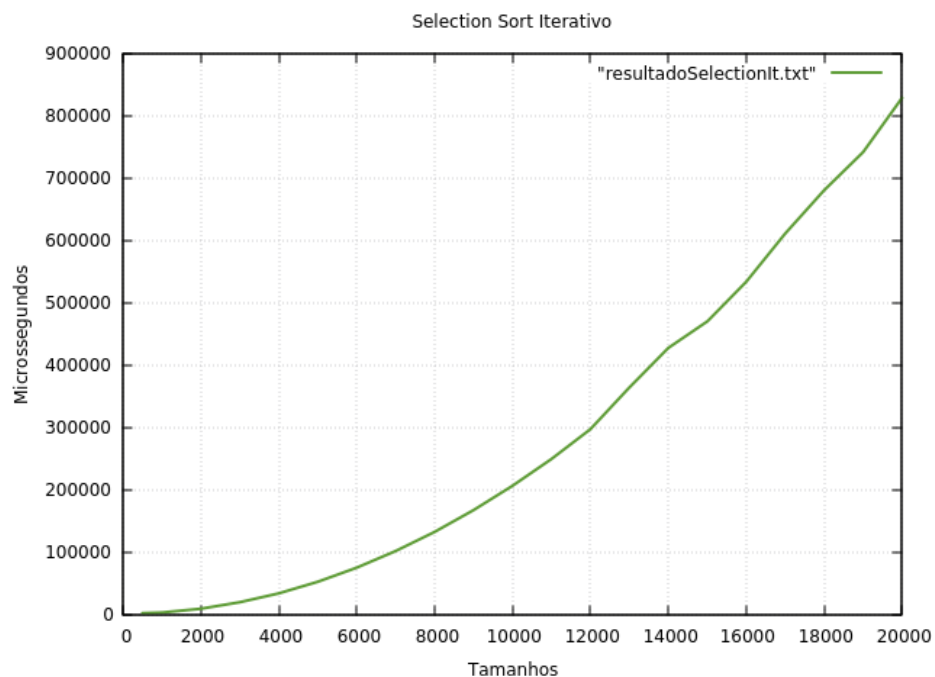


Figura 5: Selection Sort Recursivo

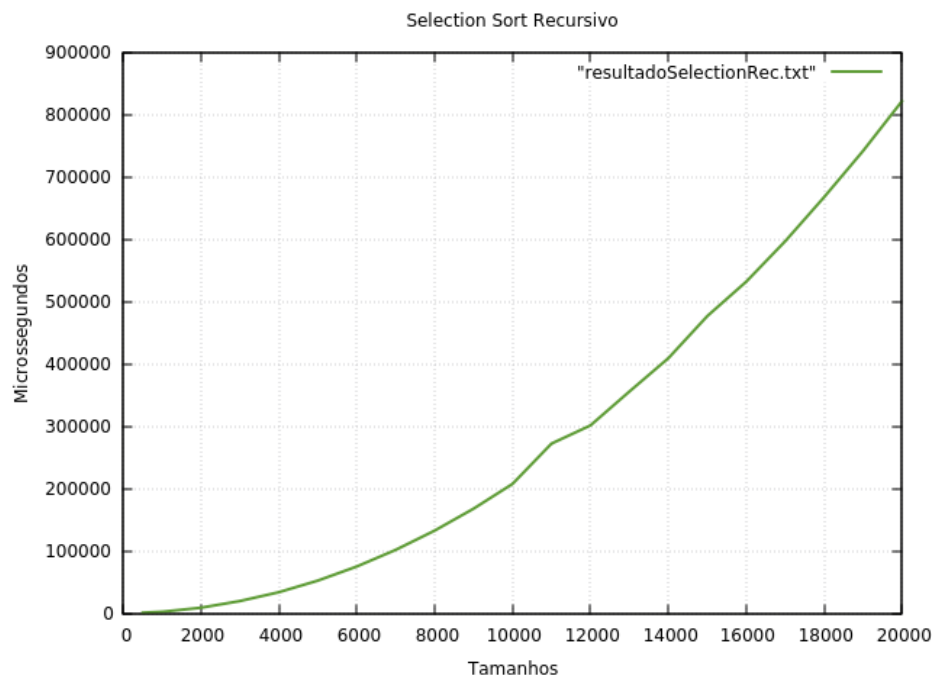
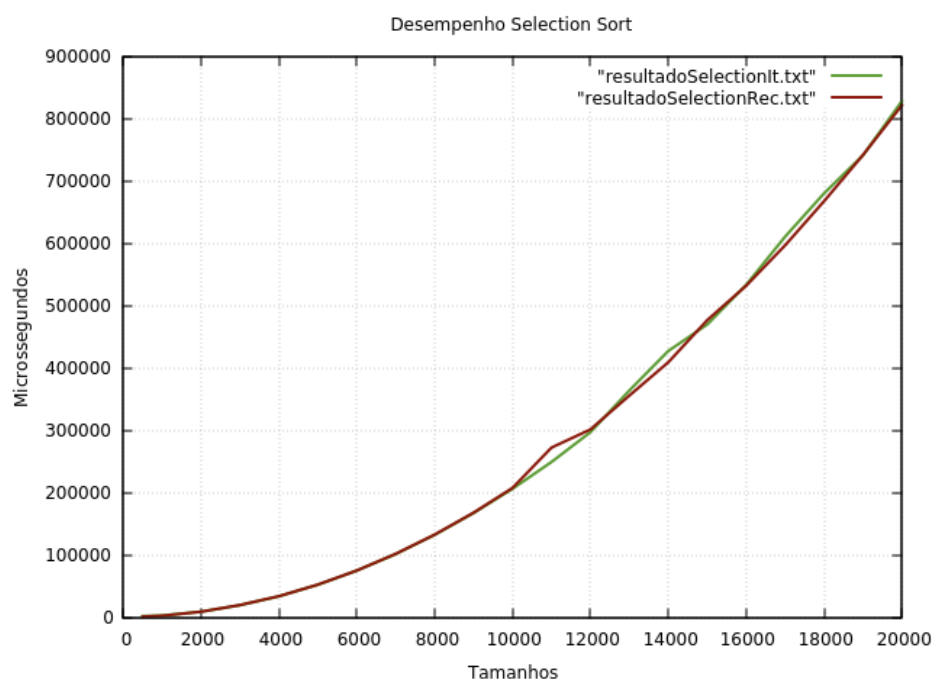


Figura 6: Comparação de desempenho

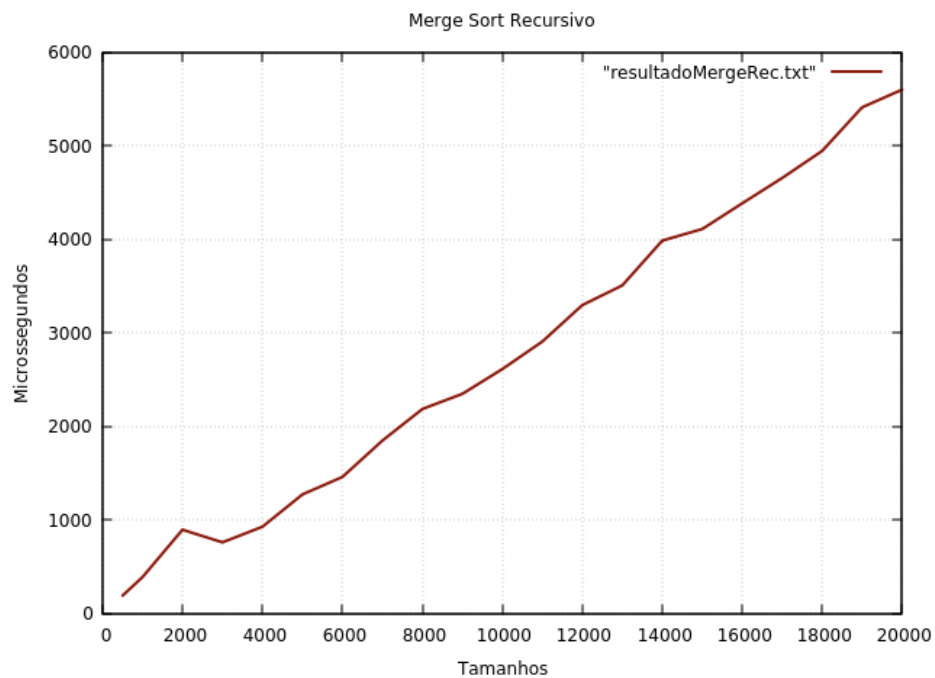


### 3 Merge Sort

#### Descrição do algoritmo

Com a estratégia de dividir para conquistar, divide o vetor completo em vetores menores para ir intercalando esses vetores menores.

Figura 7: Merge Sort Recursivo

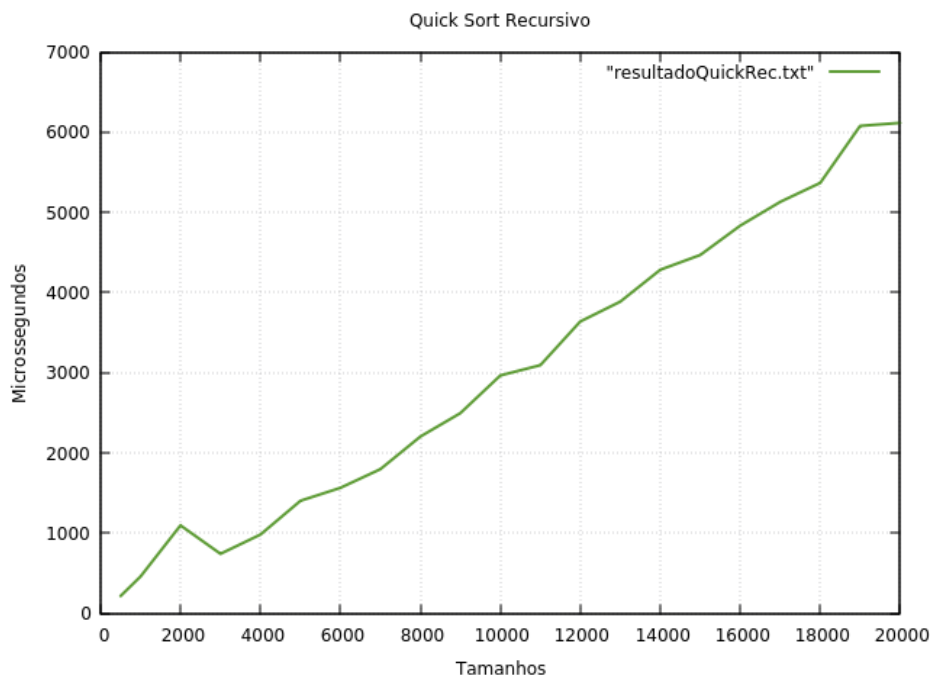


### 4 Quick Sort

#### Descrição do algoritmo

Com a mesma estratégia do merge sort de dividir para conquistar, com a diferença que é escolhido um elemento do vetor para ser o pivô, logo o pivô vai ser o separador do vetor. Os elementos menores que o pivô vão para a esquerda e os maiores para a direita, depois disso é so reorganizar os vetores que foram separados.

Figura 8: Quick Sort Recursivo



## 5 Comparação entre os algoritmos

Analisando a figura 9, com os algoritmos recursivos dá para perceber como a diferença do Quick e do Merge é grande comparado a adaptação dos algoritmos originalmente iterativos (Insertion e Selection), chegando a ser quase imperceptível o tempo de execução do Merge e do Quick.

Figura 9: Algoritmos recursivos

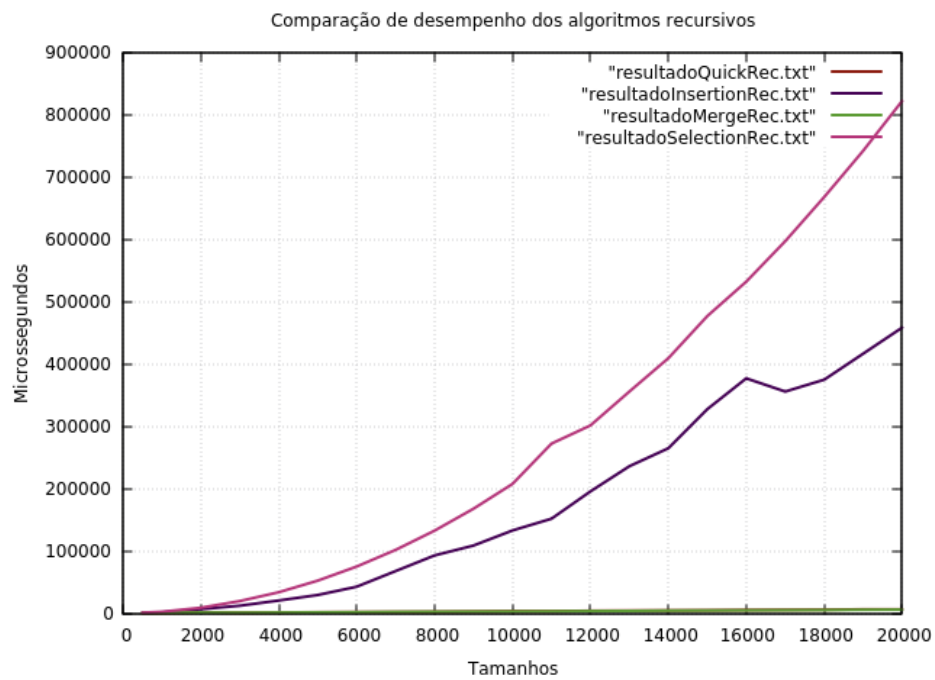
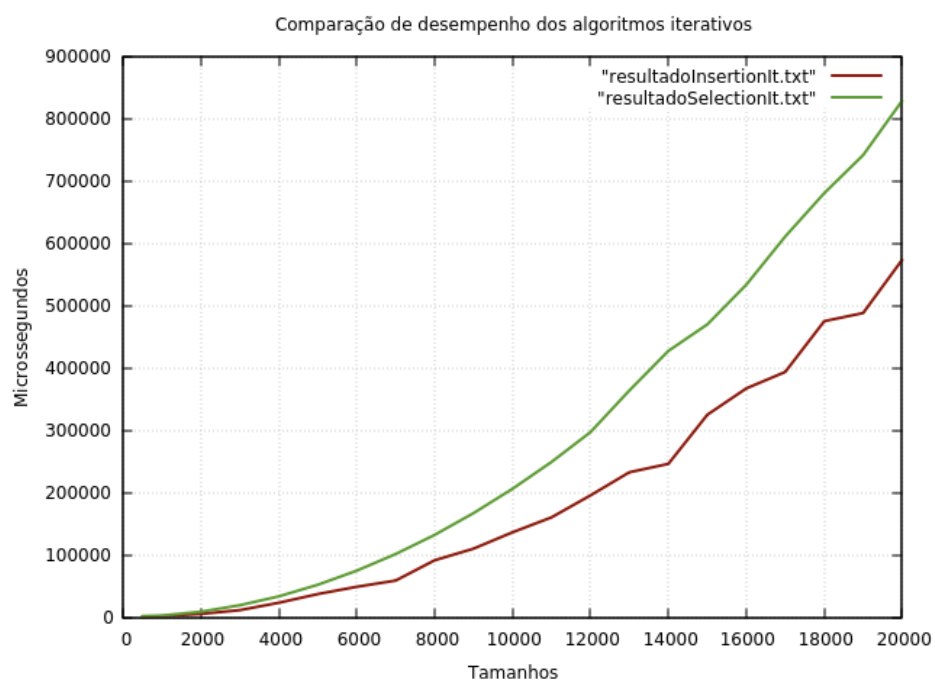


Figura 10: Algoritmos recursivos



## 6 Dificuldades na implementação

Dentre as principais dificuldades encontradas, a principal foi criar a versão iterativa de algoritmos que originalmente são recursivos, como o Quick e o Merge, não consegui criar uma lógica/estratégia para criá-los, por isso eles não foram implementados.

Outra dificuldade foi criar os arquivos .txt para conseguir plotar o gráfico(que também precisei pesquisar muito como fazer), mas usei como base o código disponibilizado pelo professor para gerar os dados e plotar os gráficos.

## 7 Bibliografia

- [Algoritmos de Ordenação Elementares - Prof. Paulo Feofiloff](#)
- Aulas da disciplina de Estrutura de dados - Prof. Atílio Gomes
- [gnuplot Installation and graph plotting tutorial on \( Linux / Ubuntu\)](#)
- [gnuplotting](#)