



UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO

Projeto e Análise de Algoritmos
Trabalho 1º Bimestre 2019

Título: Métodos de Ordenação.

Aluno: Guilherme de Oliveira Felipe

Data: 24/04/2019

1. INTRODUÇÃO

Algoritmo de ordenação em ciência da computação é um algoritmo que coloca os elementos de uma dada sequência em uma certa ordem, em outras palavras, efetua sua ordenação completa ou parcial. As ordens mais usadas são a numérica e a lexicográfica. Neste trabalho o objetivo foi analisar os algoritmos de ordenação de 20 bases numéricas diferentes, na qual a primeira contém 100 valores e a última 2.000.000 valores. O número de comparações e swap está implementado no código, portanto estudaremos apenas o comportamento temporal neste relatório.

2. OBJETIVOS

Aplicar e corroborar conceitos adquiridos com relação aos métodos de ordenação.

3. MATERIAL UTILIZADO

- Sistema Operacional: Ubuntu 18.04.2 LTS baseado em 64-bit.
- Memória RAM: 15,6 GB.
- Processador: AMD FX-8150 Eight-Core Processor x8.
- IDE: Clion.

4. EXECUÇÃO DO CÓDIGO

Para executar o algoritmo o usuário deverá especificar a pasta destino das bases de dados (linha 37 do código) , na qual as bases deverão estar nomeadas de 0 a 20. Após isto, apenas relaxar enquanto os algoritmos são executados e tendo os resultados exibidos no próprio terminal da aplicação.

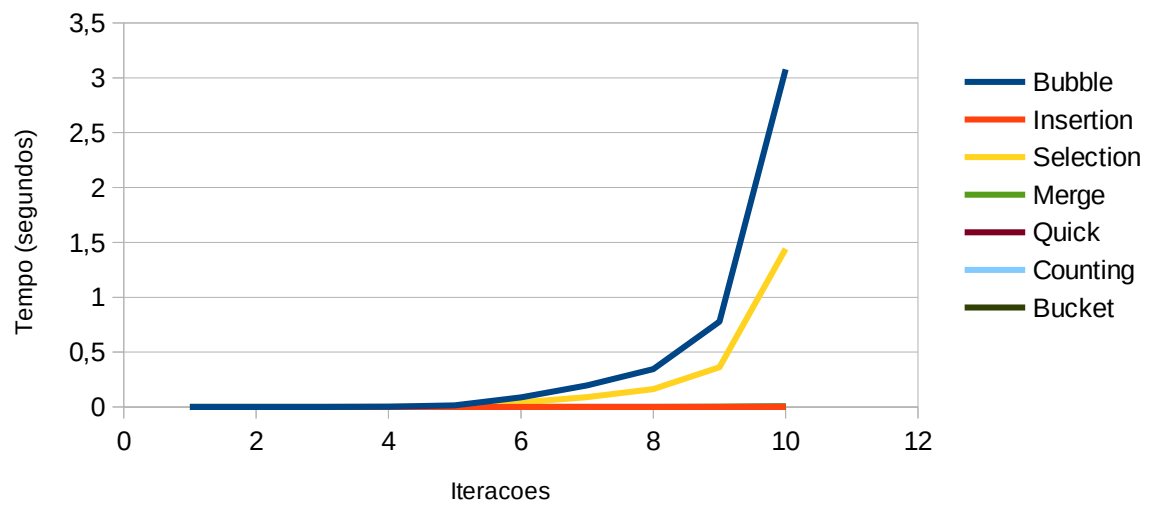
5. RESULTADOS E DISCUSSÃO

Apresentar, em ordem cronológica e lógica, os resultados, assim como analisá-los com as observações e comentários pertinentes.

Faremos a primeira análise nos algoritmos utilizando valores ordenados. Para melhor visualização, dividiremos em 3 partes sendo elas da iteração 1ª até a 10ª, seguido da 11ª até 15ª e por fim da 16ª até a 20ª. Não obstante, daremos também um zoom separando os algoritmos da seguinte maneira: bubble, insertion e selection em uma figura e noutra os restantes. As figura apresentam o número da iteração no eixo X e o tempo em segundos no eixo Y.

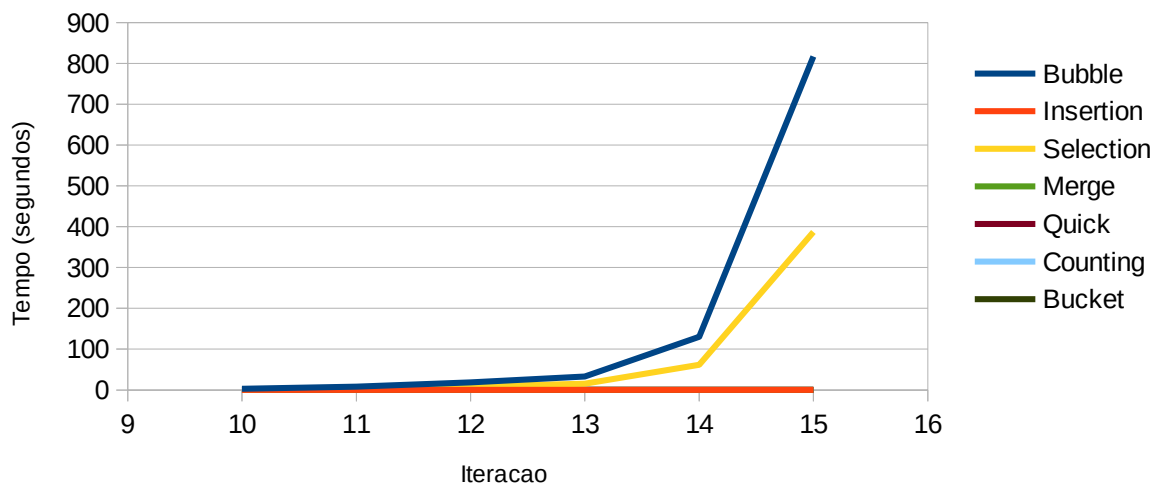
Ordenados

1/3



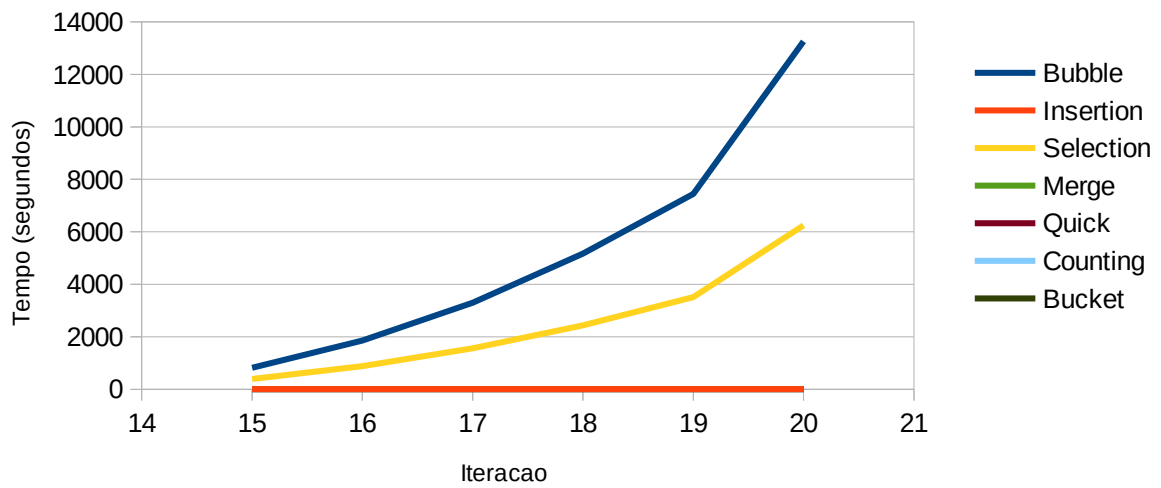
Ordenados

2/3



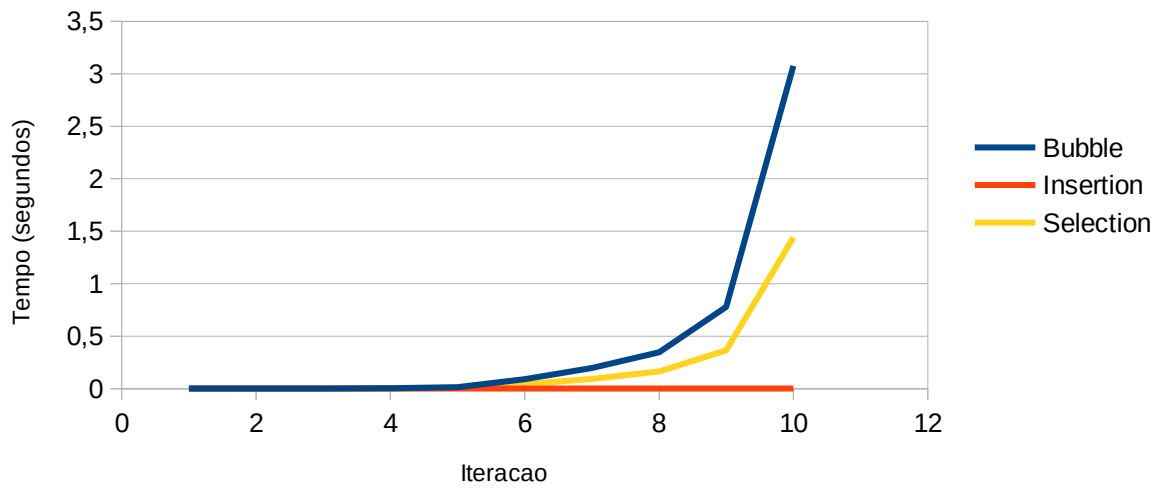
Ordenados

3/3



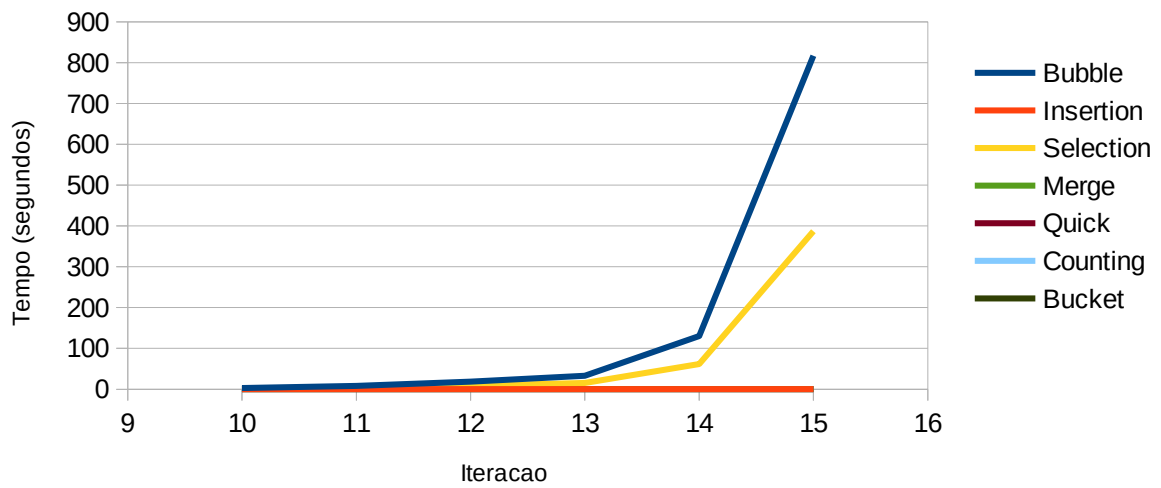
Zoom Ordenados

1/6



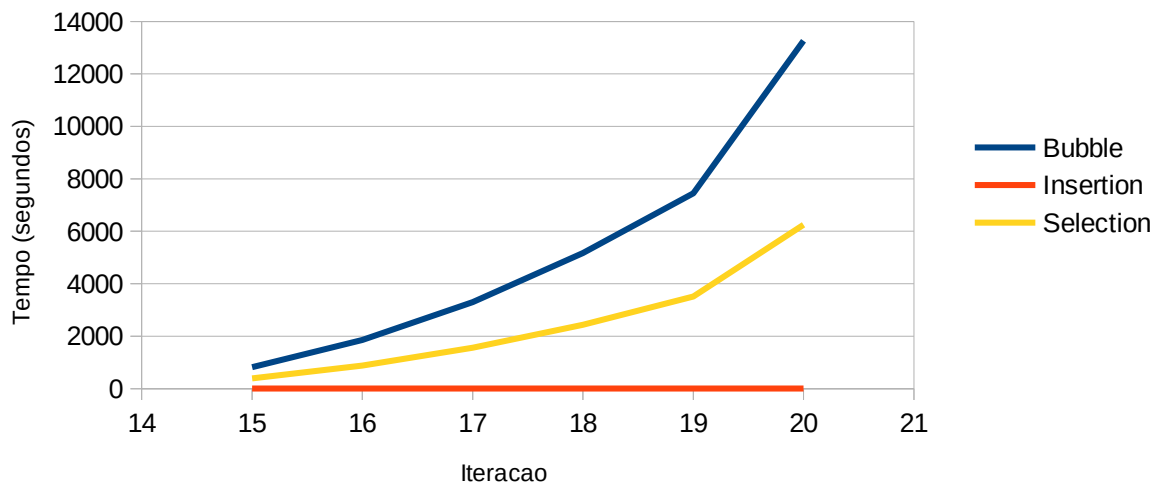
Ordenados

2/3



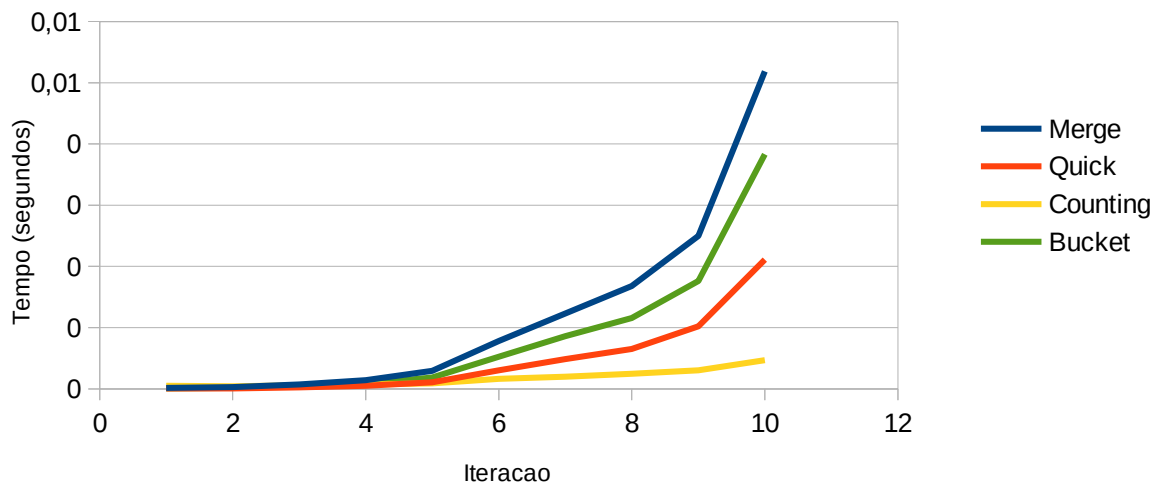
Zoom Ordenados

3/6



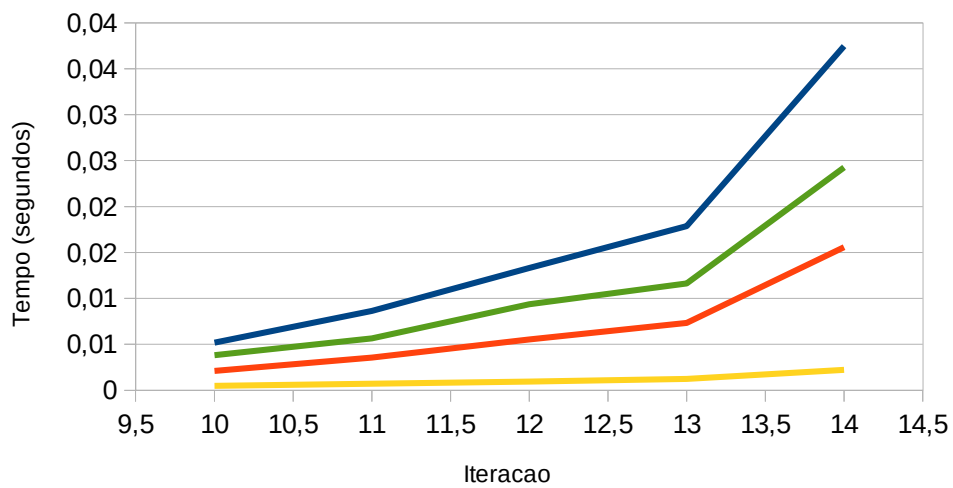
Zoom Ordenados

4/6



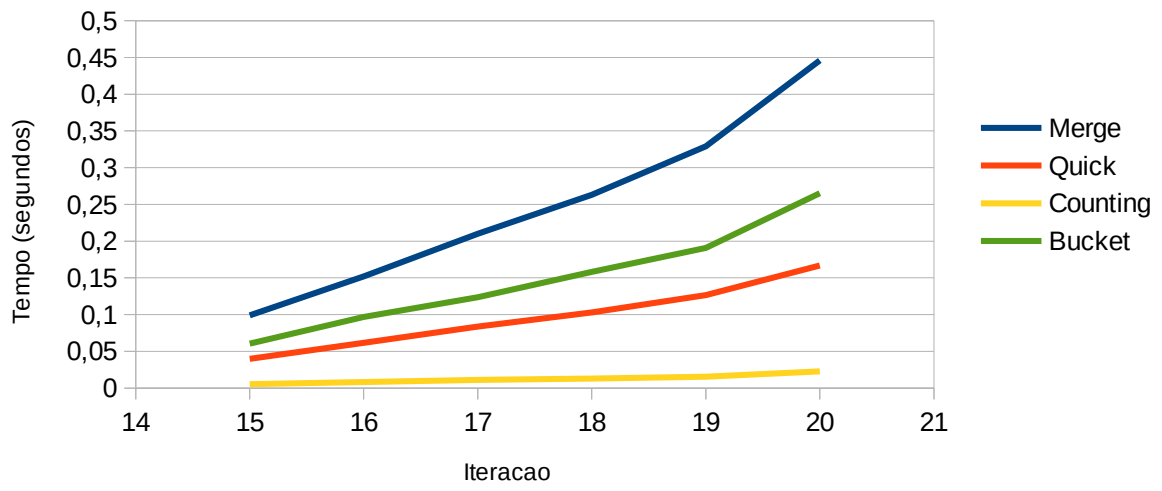
Zoom Ordenados

5/6



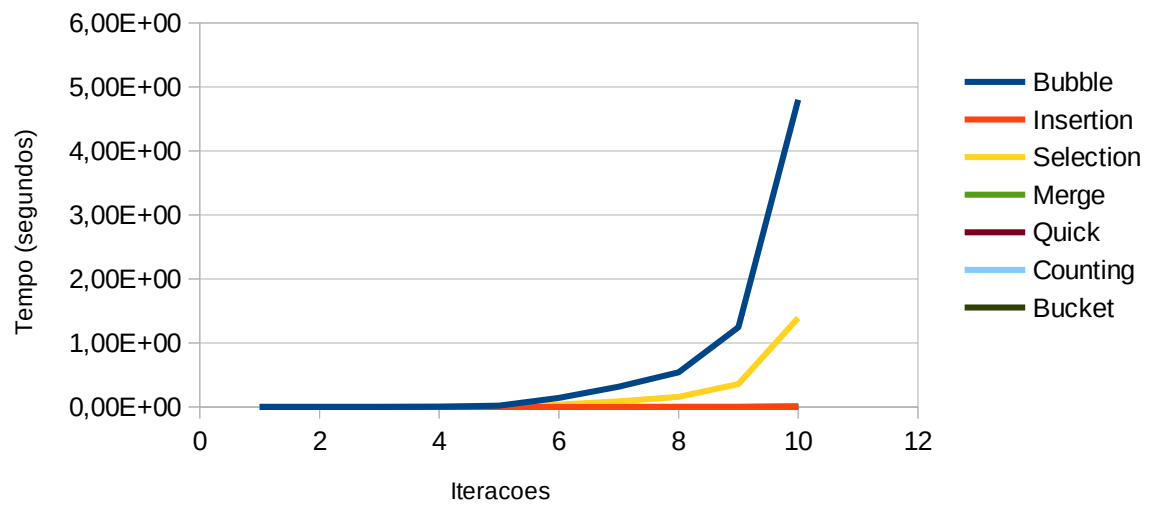
Zoom Ordenados

6/6



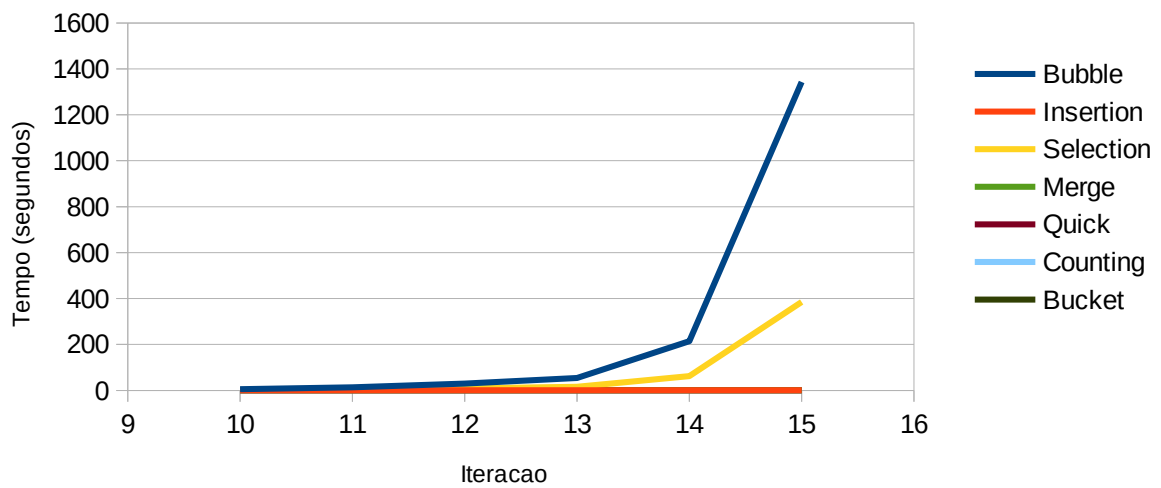
Aleatórios

1/3



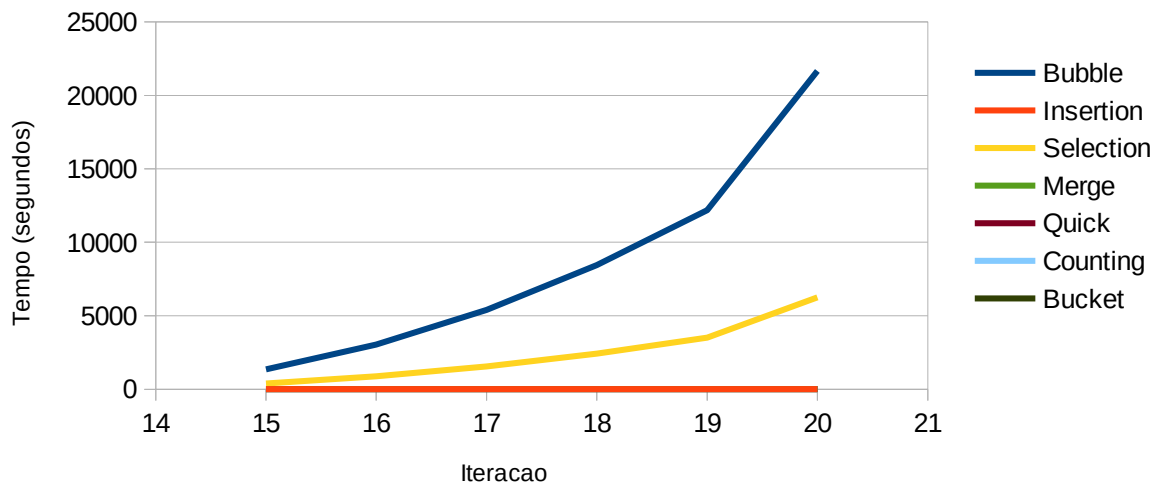
Aleatórios

2/3



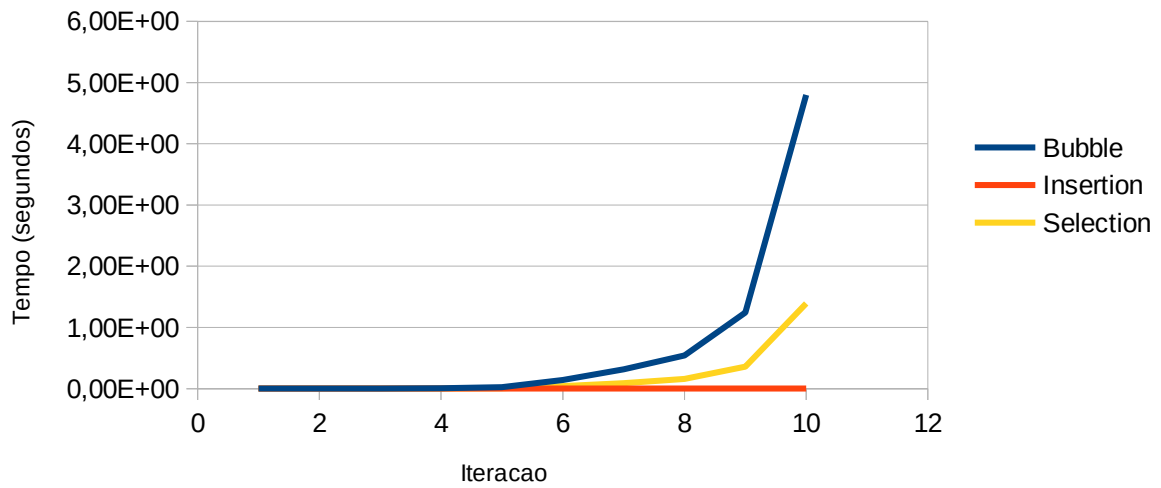
Aleatórios

3/3



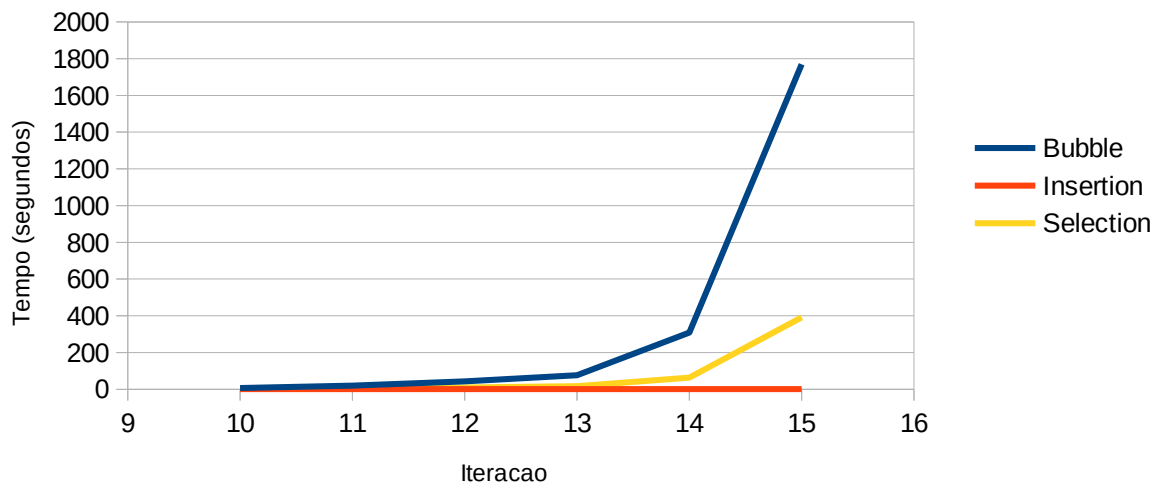
Zoom Aleatórios

1/6



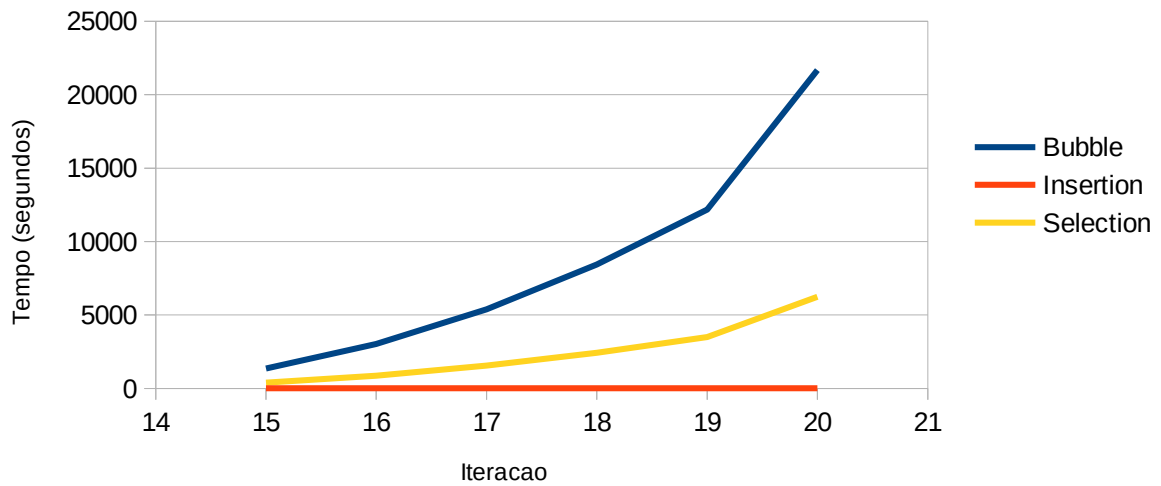
Zoom Aleatórios

2/6



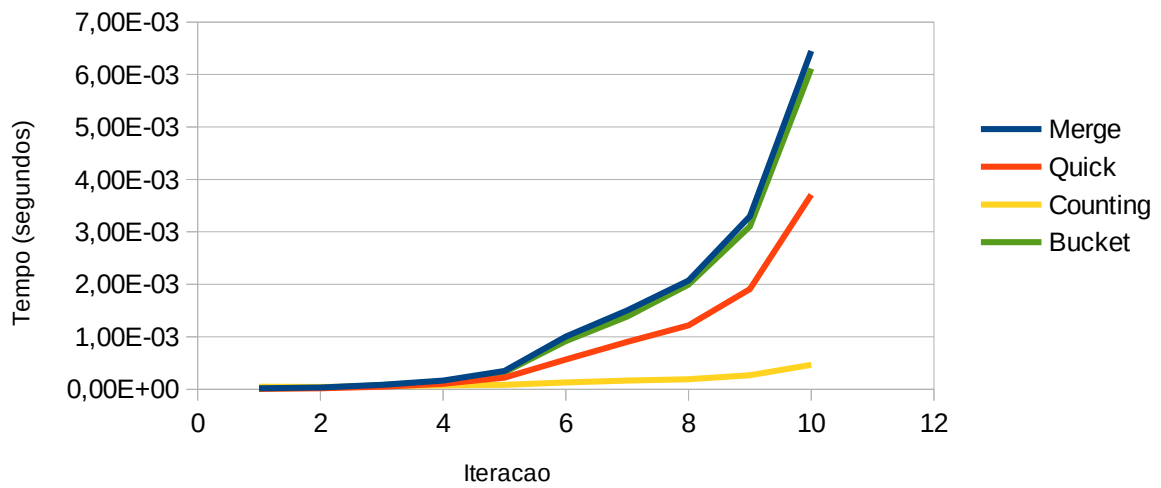
Zoom Aleatórios

3/6



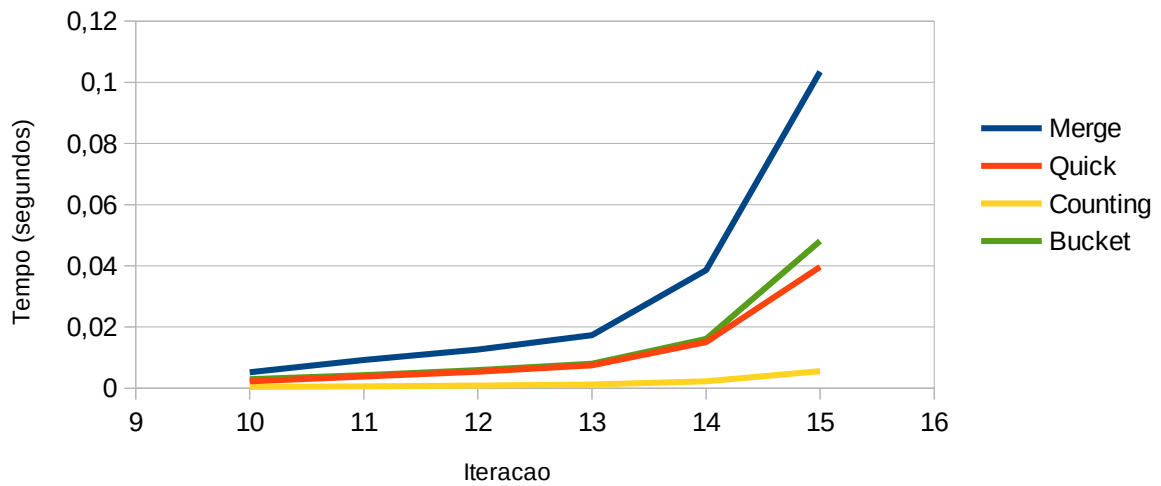
Zoom Aleatórios

4/6



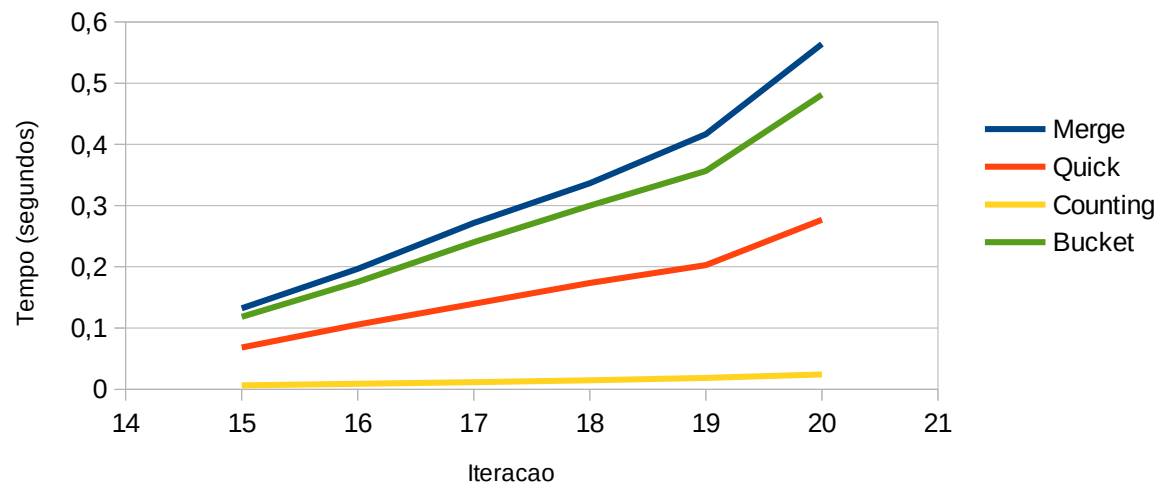
Zoom Aleatórios

5/6



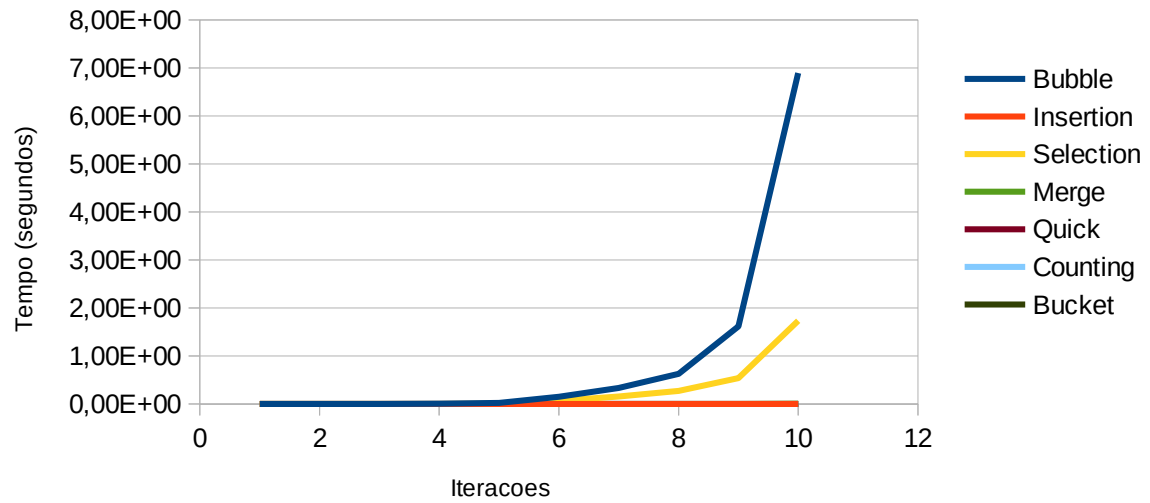
Zoom Aleatórios

6/6



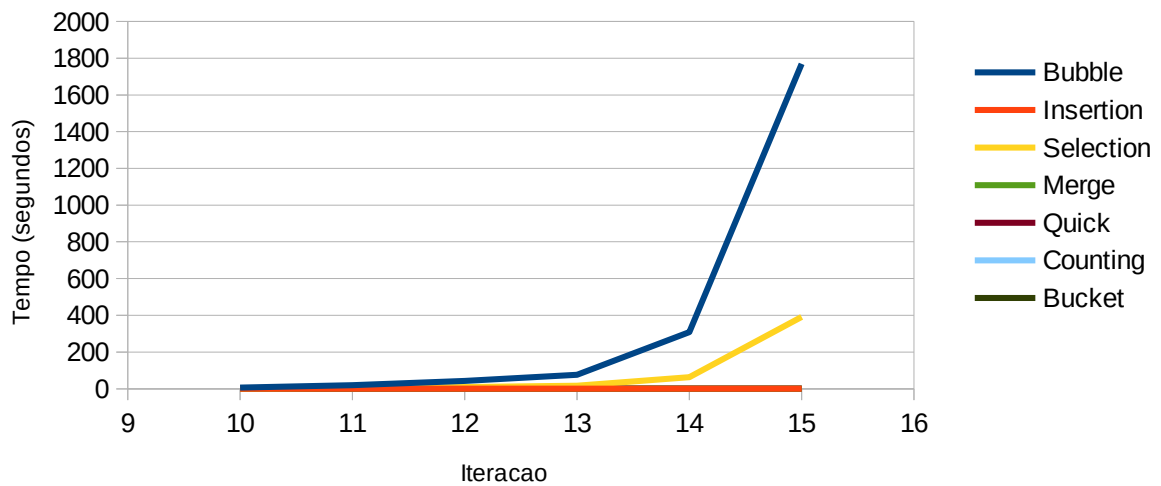
Decrescente

1/3



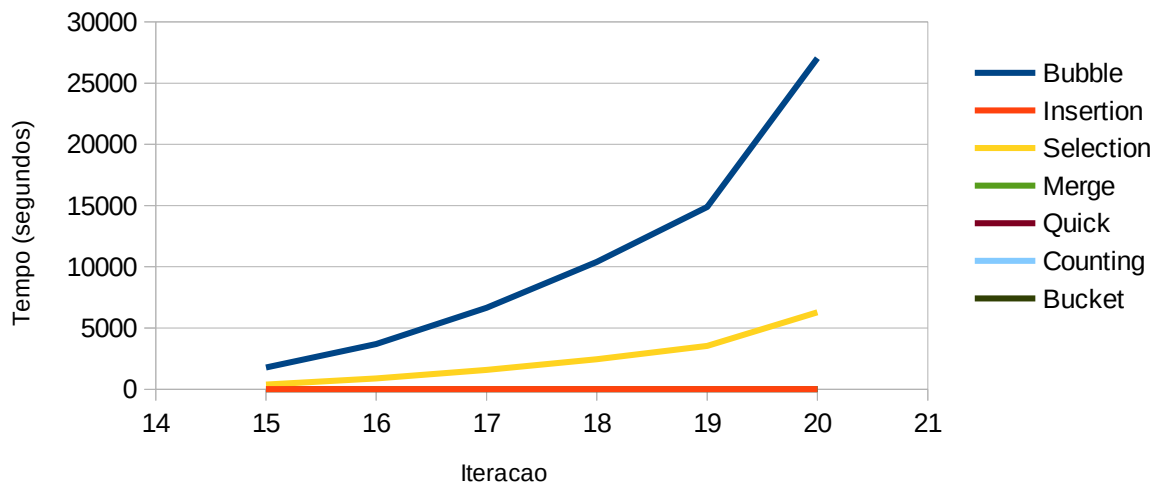
Decrescente

2/3



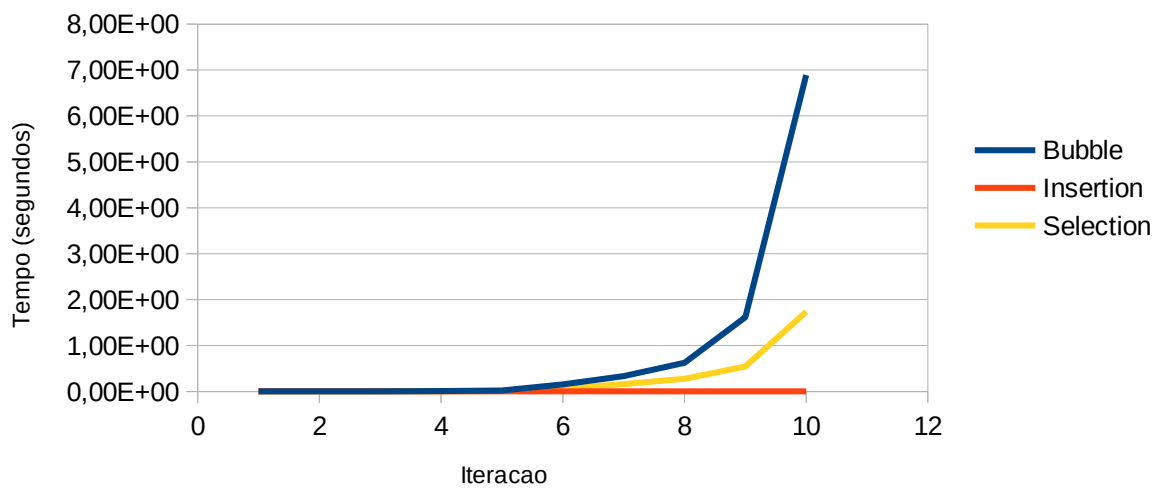
Decrescente

3/3



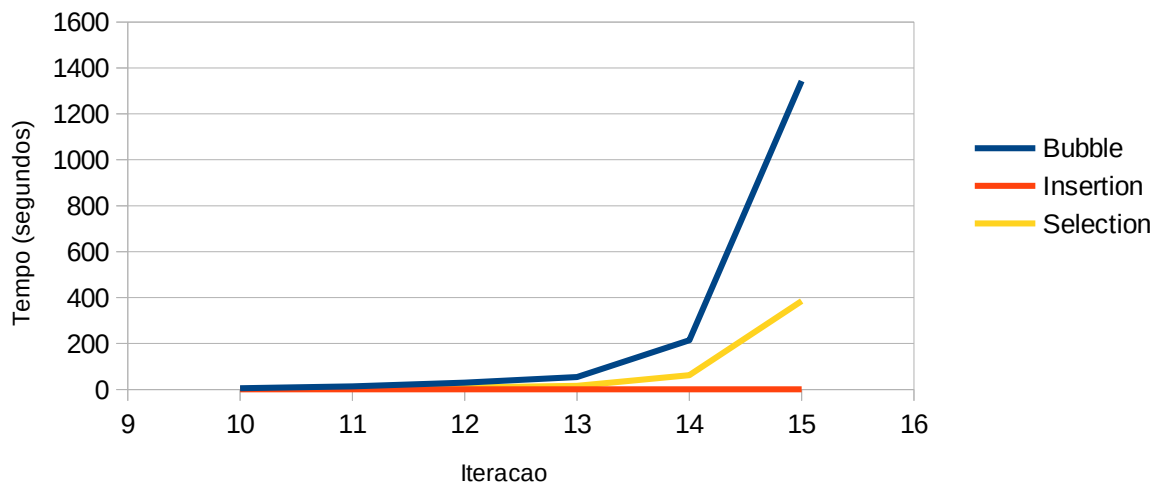
Zoom Decrescente

1/6



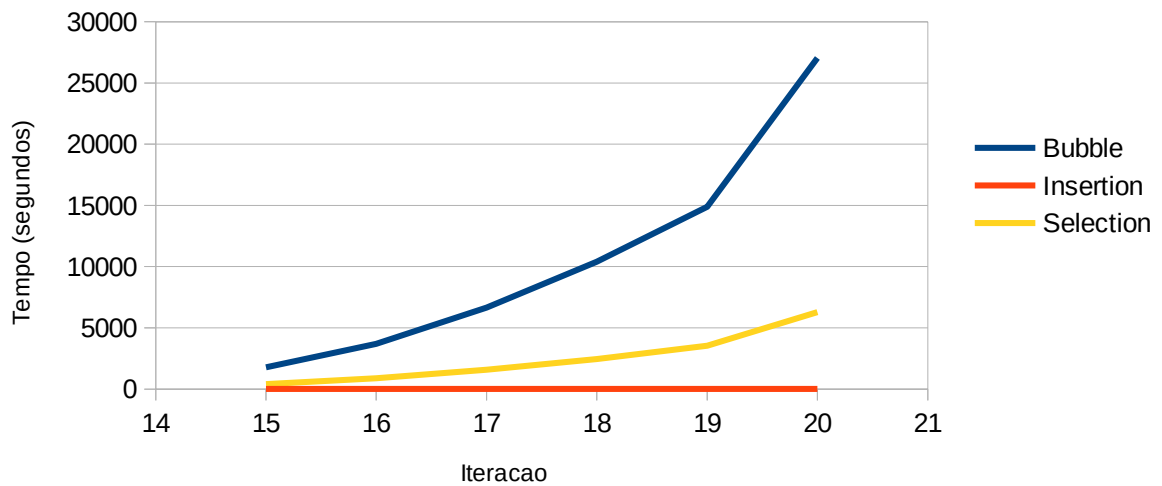
Zoom Decrescente

2/6



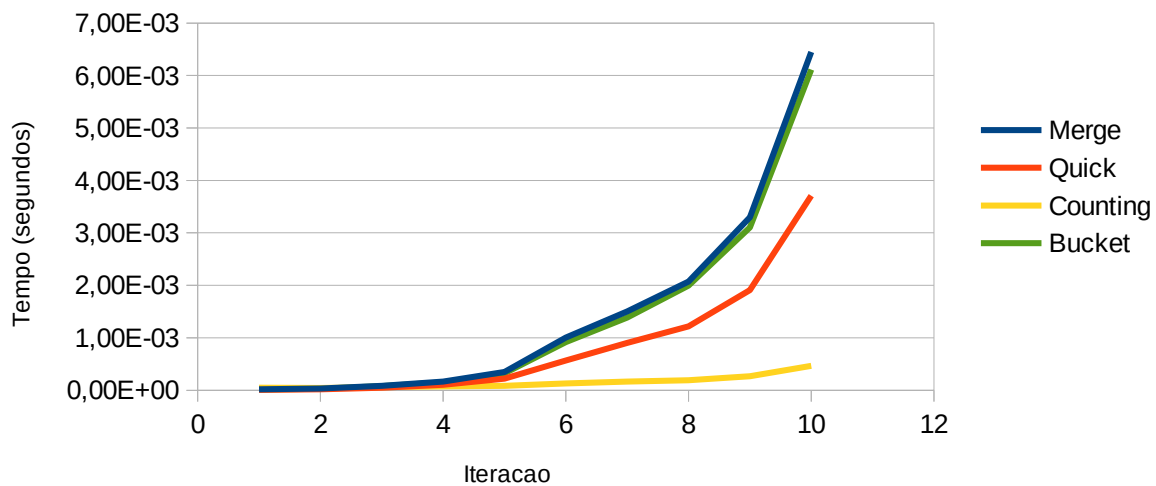
Zoom Decrescente

3/6



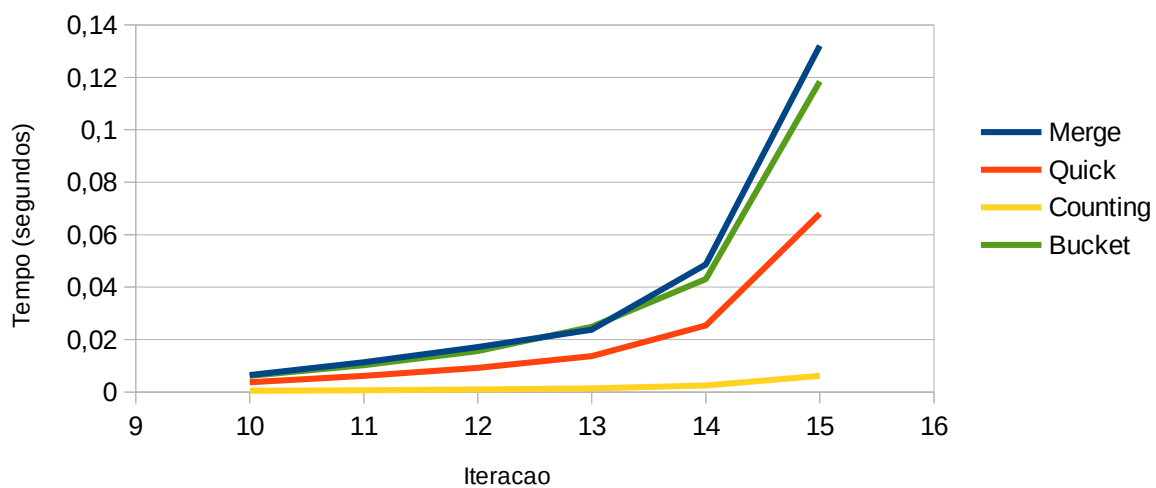
Zoom Decrescente

4/6



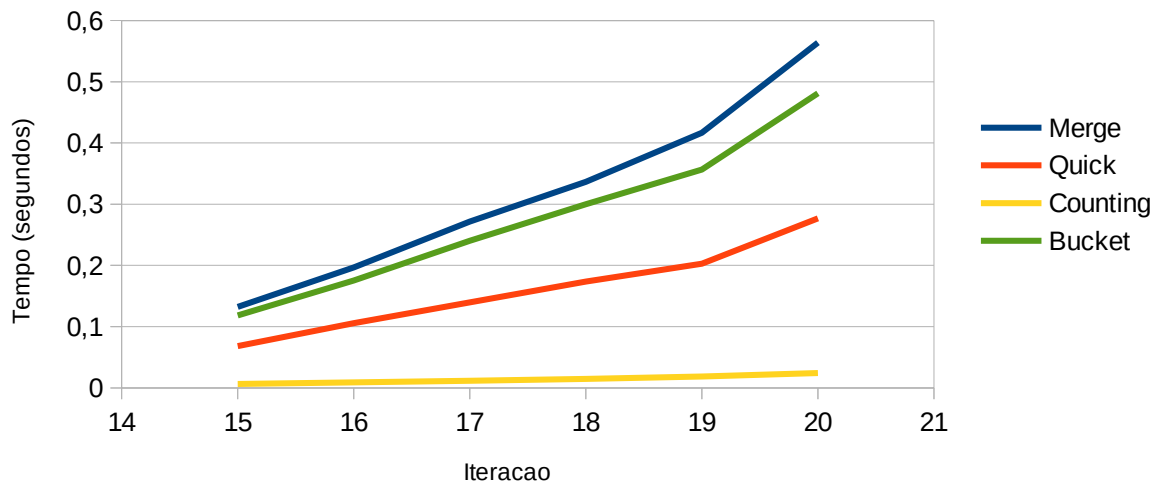
Zoom Decrescente

5/6



Zoom Decrescente

6/6



6. CONCLUSÕES

A partir da observação dos gráficos, notamos que os algoritmos quadráticos crescem de forma muito mais ascentuada que os demais, portanto são muito mais custosos. Para todos os casos o algoritmo Counting se mostrou mais veloz, apesar do problema de alocação de memória. Seguido então pelo Quick, Bucket e Merge.