

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

CAMPUS CHAPECÓ

Ciência da Computação

Disciplina: Pesquisa e ordenação de dados

Trabalho: Comparação de métodos de ordenação linearítmicos

Alisson Luan de Lima Peloso

Chapecó - SC

2020

# **Tabelas**:

## Merge sort

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ordenada** | **Inversamente Ordenada** | **Aleatório** |
|  | **Merge sort** | | |
| **Tempo (ms)** | | |
| **10.000** | 1,641 | 3,138 | 5,403 |
| **50.000** | 7,288 | 8,180 | 11,482 |
| **100.000** | 14,285 | 13,260 | 23,998 |

## Quick sort

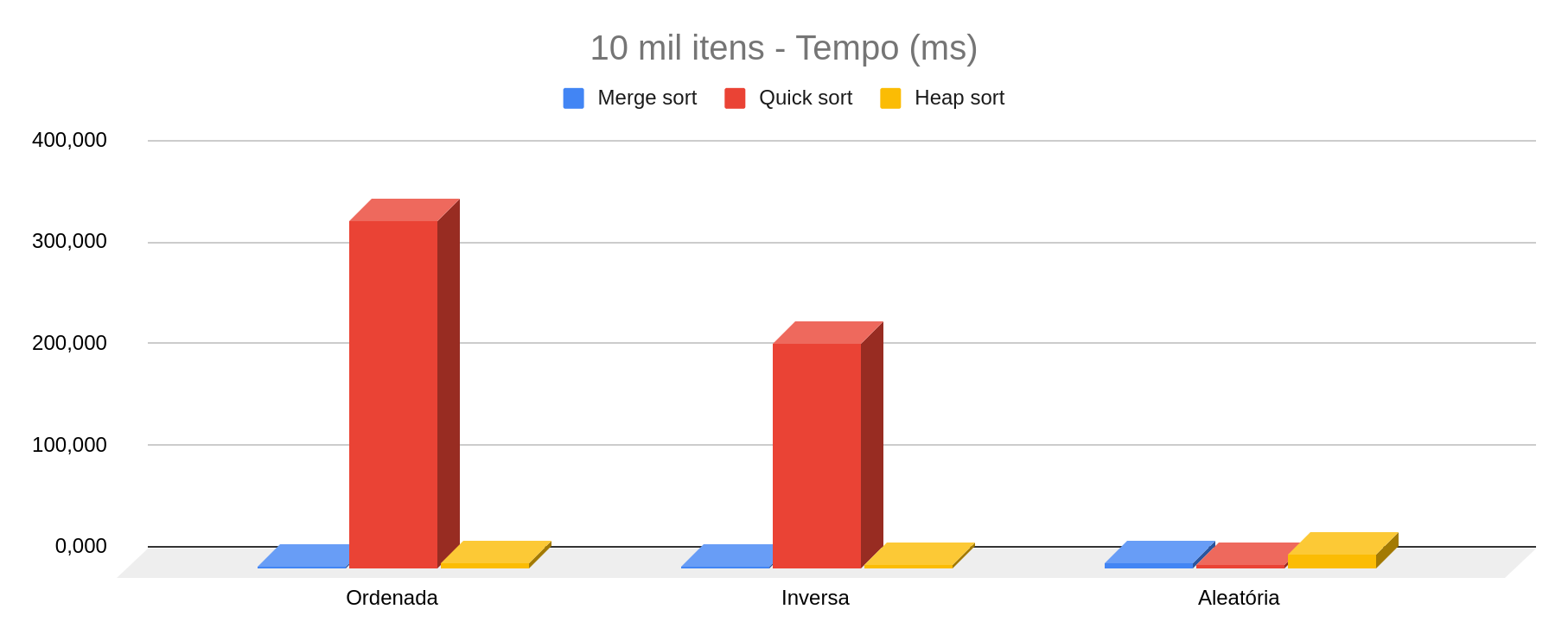
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ordenada** | **Inversamente Ordenada** | **Aleatório** |
|  | **Quick sort** | | |
| **Tempo (ms)** | | |
| **10.000** | 342,334 | 221,062 | 4,533 |
| **50.000** | 8.144,809 | 5.151,932 | 10,781 |
| **100.000** | 31.703,258 | 21.952,376 | 19,094 |

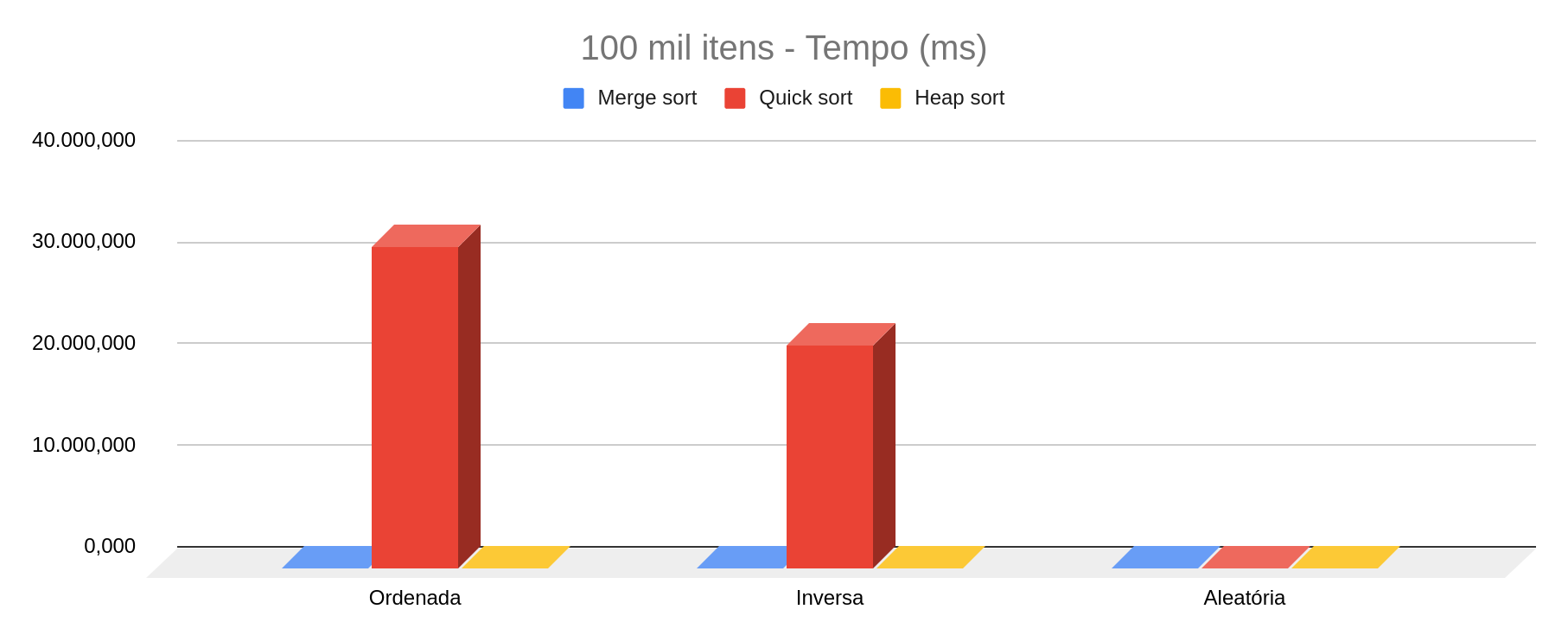
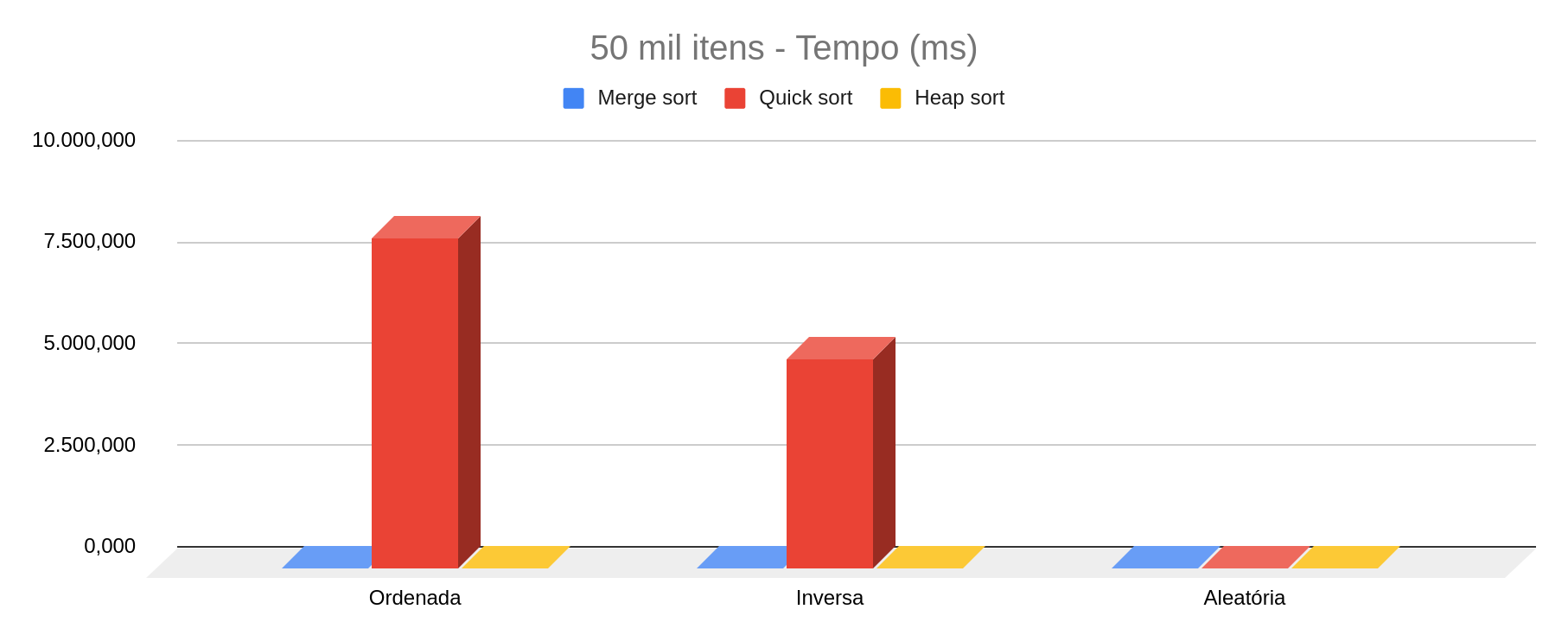
## Heap sort

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ordenada** | **Inversamente Ordenada** | **Aleatório** |
|  | **Heap sort** | | |
| **Tempo (ms)** | | |
| **10.000** | 5,963 | 3,885 | 13,804 |
| **50.000** | 12,648 | 12,568 | 17,340 |
| **100.000** | 27,543 | 25,226 | 32,653 |

# Gráficos Comparativos:

## Tempo de Execução





## 

# Conclusão:

Ao realizar a comparação com os gráficos é possível notar que os tempos de execução se mantém na mesma proporção ao alterarmos a quantidade de itens a serem ordenados.

Observando os gráficos, notamos que o Merge e o Quick sort não possuem uma diferença tão gritante nos tempos de execução em relação ao estado inicial da lista (ordenada, inversamente ordenada, aleatória) quanto o Quick sort.

No Merge sort, a lista ordenada teve menor tempo de execução, na lista aleatória teve o maior tempo e a mediana ficou para a lista inversamente ordenada. Isso predominou nos três casos: ordenada, inversa e aleatória.

Já no Heap sort, a lista inversa teve o menor tempo de execução, seguida da lista ordenada. A aleatória ficou com o maior tempo.

O caso mais peculiar foi o do Quick sort. A listas ordenada e inversa executaram em um tempo muito superior em comparação com os outros métodos. Porém, no caso em que a lista inicial está em ordem aleatória, o Quick sort ordenou a lista em menos tempo que os outros métodos linearítmicos.

Por fim, entendemos que o desempenho dos algoritmos de ordenação dependem da forma inicial da lista, e para cada caso há um algoritmo que terá melhor desempenho. Em comparação com os métodos de ordenação simples, o tempo de execução nos diferentes tamanhos de lista não gera uma diferença exponencial. Isso torna os métodos de ordenação linearítmicos mais eficientes para listas maiores.

## Hardware Utilizado

**Processador**: Intel© Core™ i5-7200U CPU @ 2.50GHz × 2 with Turbo Boost up to 3.1Ghz

**Memória RAM**: 8 GB DDR4 Memory

# 