

# Relatório de CES-11

Levi de Lima

Setembro 2022

## 1 Teste dos dois segundos

A tabela a seguir mostra, para cada algoritmo, o tamanho máximo da entrada que o programa roda em dois segundos.

| Algoritmo    | Tamanho da entrada |
|--------------|--------------------|
| Bubble       | 12000              |
| Merge        | 150000             |
| Merge Static | 1600000            |
| Quick        | 1900000            |

Com análise da tabela é possível perceber que o Merge Sort utilizando *static* é mais eficiente que aquele que não usa. Doravante, toda vez que esse algoritmo for mencionado assume-se o mais eficiente.

## 2 Análise

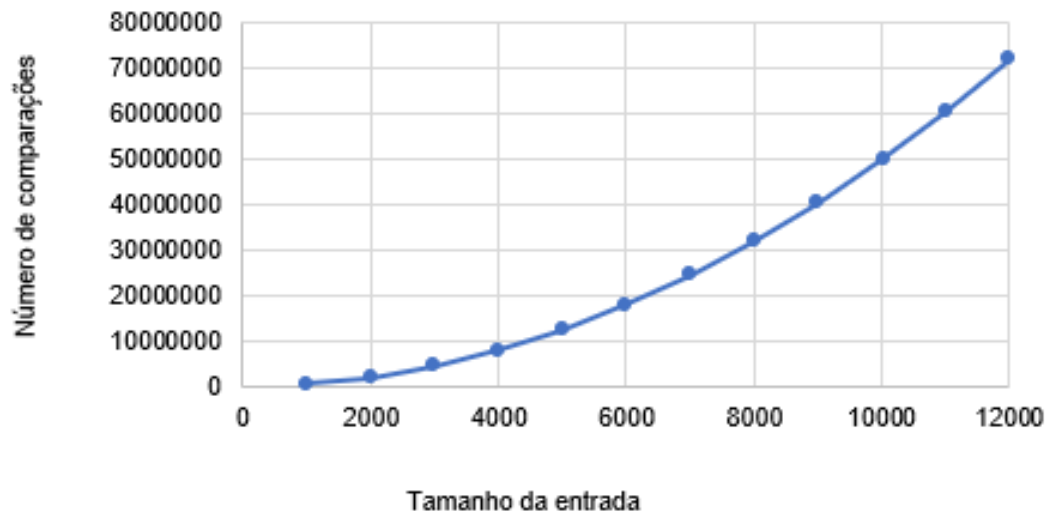
### 2.1 Bubble Sort

O tamanho máximo do vetor que o Bubble Sort consegue rodar em dois segundos é 12.000. Abaixo há uma tabela que compara o tamanho da entrada, o número de comparações e o tempo gasto pelo algoritmo.

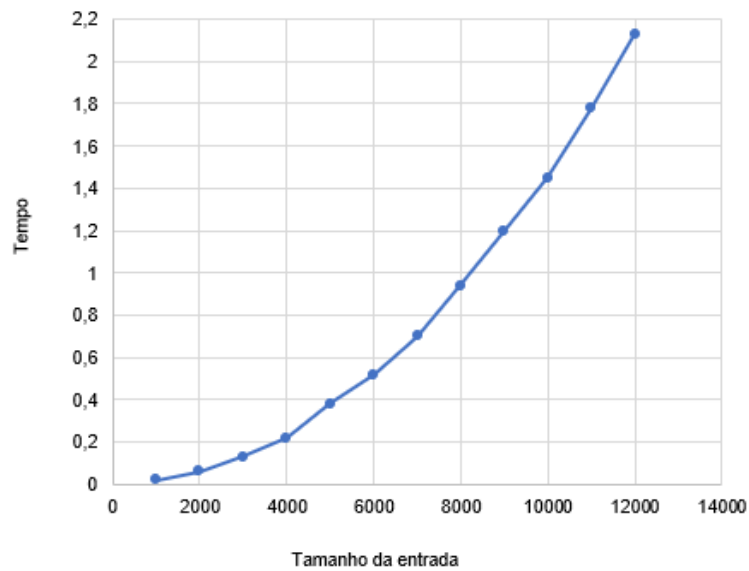
| Tamanho da Entrada | Número de Comparações | Tempo |
|--------------------|-----------------------|-------|
| 1000               | 499500                | 0,02  |
| 2000               | 1999000               | 0,06  |
| 3000               | 4498500               | 0,13  |
| 4000               | 7998000               | 0,22  |
| 5000               | 12497500              | 0,38  |
| 6000               | 17997000              | 0,52  |
| 7000               | 24496500              | 0,7   |
| 8000               | 31996000              | 0,94  |
| 9000               | 40495500              | 1,2   |
| 10000              | 49995000              | 1,45  |
| 11000              | 60494500              | 1,78  |
| 12000              | 71994000              | 2,13  |

A seguir, estão representados dois gráficos, o primeiro descreve a variação do número de comparações em função do tamanho da entrada e o segundo o tempo *versus* o tamanho da entrada.

**Tamanho da entrada X Número de comparações**



**Tamanho da entrada X Tempo**



A análise dos gráficos e dos formatos das curvas nos permite concluir que os resultados da relação entre o tempo gasto e o número de comparações estão coerentes com o esperado ( $O(n^2)$ ).

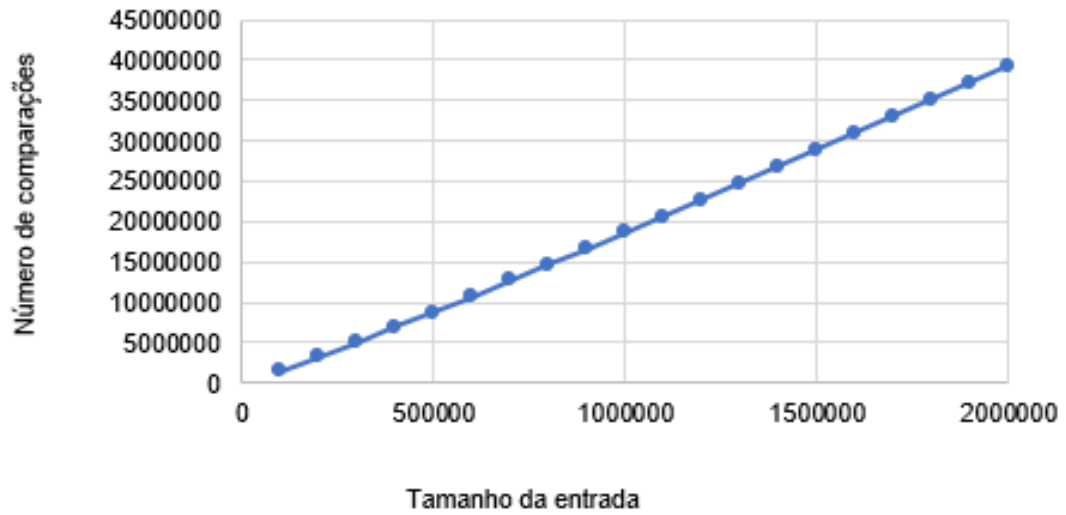
## 2.2 Merge Sort

O tamanho máximo do vetor que o Merge Sort consegue rodar em dois segundos é 1.600.000. Abaixo há uma tabela que compara o tamanho da entrada, o número de comparações e o tempo gasto pelo algoritmo.

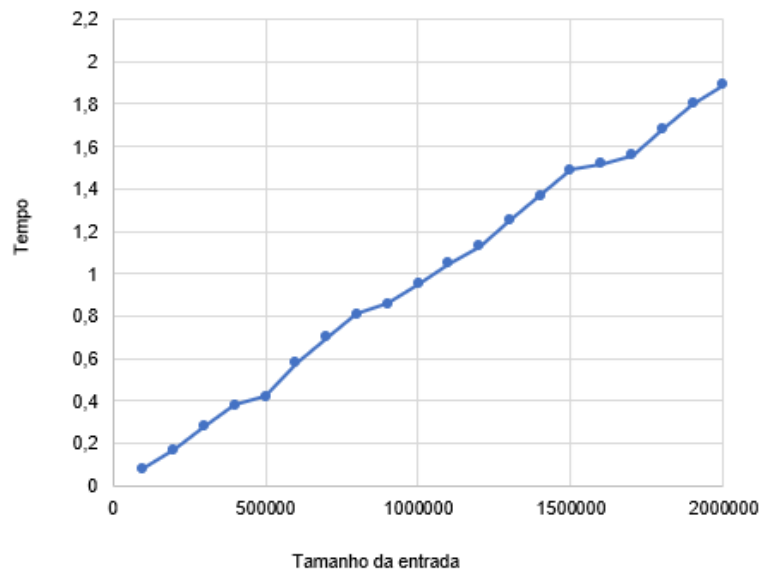
| Tamanho da Entrada | Número de Comparações | Tempo |
|--------------------|-----------------------|-------|
| 100000             | 1536482               | 0,08  |
| 200000             | 3272366               | 0,17  |
| 300000             | 5084057               | 0,28  |
| 400000             | 6945133               | 0,38  |
| 500000             | 8836479               | 0,42  |
| 600000             | 10768748              | 0,58  |
| 700000             | 12723309              | 0,7   |
| 800000             | 14690641              | 0,81  |
| 900000             | 16676918              | 0,86  |
| 1000000            | 18672862              | 0,95  |
| 1100000            | 20694589              | 1,05  |
| 1200000            | 22738506              | 1,13  |
| 1300000            | 24787466              | 1,25  |
| 1400000            | 26846426              | 1,37  |
| 1500000            | 28910334              | 1,49  |
| 1600000            | 30980469              | 1,52  |
| 1700000            | 33067279              | 1,56  |
| 1800000            | 35155979              | 1,68  |
| 1900000            | 37248883              | 1,8   |
| 2000000            | 39351054              | 1,89  |

A seguir, estão representados dois gráficos, o primeiro descreve a variação do número de comparações em função do tamanho da entrada e o segundo o tempo *versus* o tamanho da entrada.

**Tamanho da entrada X Número de comparações**



**Tamanho da entrada X Tempo**



A análise dos gráficos e dos formatos das curvas nos permite concluir que os resultados da relação entre o tempo gasto e o número de comparações estão coerentes com o esperado ( $O(n \log n)$ ).

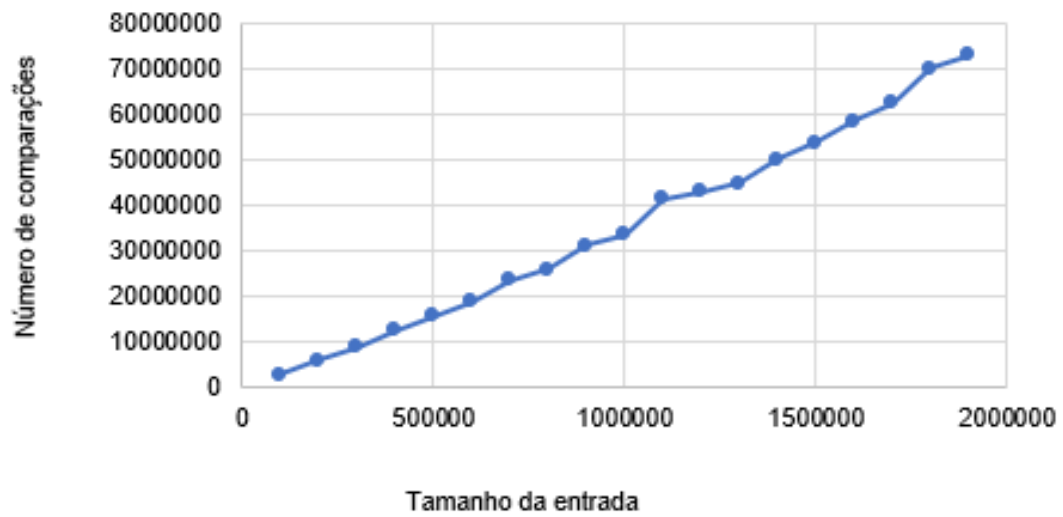
### 2.3 Quick Sort

O tamanho máximo do vetor que o Quick Sort consegue rodar em dois segundos é 1.900.000. Abaixo há uma tabela que compara o tamanho da entrada, o número de comparações e o tempo gasto pelo algoritmo.

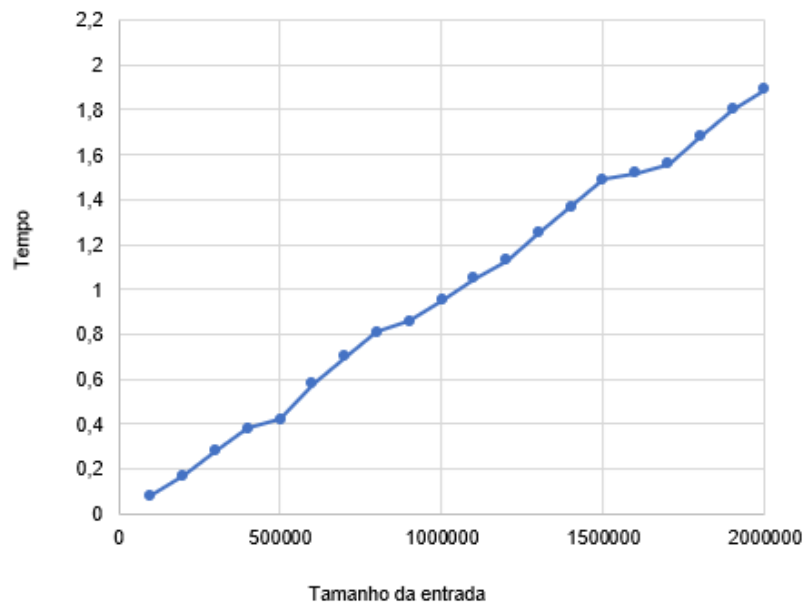
| Tamanho da Entrada | Número de Comparações | Tempo |
|--------------------|-----------------------|-------|
| 100000             | 2702930               | 0,05  |
| 200000             | 5755499               | 0,14  |
| 300000             | 8871253               | 0,19  |
| 400000             | 12392550              | 0,27  |
| 500000             | 15686970              | 0,33  |
| 600000             | 18785698              | 0,41  |
| 700000             | 23413523              | 0,53  |
| 800000             | 25857983              | 0,78  |
| 900000             | 31103363              | 0,87  |
| 1000000            | 33592379              | 0.97  |
| 1100000            | 41373530              | 1.08  |
| 1200000            | 42948497              | 1.22  |
| 1300000            | 44975018              | 1.31  |
| 1400000            | 50176278              | 1.44  |
| 1500000            | 53812583              | 1.53  |
| 1600000            | 58579104              | 1.66  |
| 1700000            | 62409351              | 1.78  |
| 1800000            | 70306552              | 1.92  |
| 1900000            | 73074923              | 2.05  |
| 2000000            | 39351054              | 1,89  |

A seguir, estão representados dois gráficos, o primeiro descreve a variação do número de comparações em função do tamanho da entrada e o segundo o tempo *versus* o tamanho da entrada.

**Tamanho da entrada X Número de comparações**



**Tamanho da entrada X Tempo**



A análise dos gráficos e dos formatos das curvas nos permite concluir que os resultados da relação entre o tempo gasto e o número de comparações estão coerentes com o esperado ( $O(n \log n)$ ).

### 3 Informações

Foram utilizados os seguintes programas para a confecção deste relatório:

1. Lab3\_Levi\_Gurgel\_de\_Lima.cpp (autoria própria)
2. Lab3\_Levi\_Gurgel\_de\_Lima\_relatório.cpp (autoria própria)
3. Lab3\_gerador\_strings\_ACGT.cpp (fornecido no classroom)