

Análise dos algoritmos

-Análise dos QuickSorts

-Tabelas

As tabelas a seguir foram criadas usando os dados retornados por cada tipo de QuickSort com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.

| Tipo QuickSort (N=1000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 4,907 | 4,859 | 0.011 |
| QuickSort Mediana de 3 valores | 5,726 | 6,058 | 0.179 |
| QuickSort Mediana de 5 valores | 4,748 | 4,927 | 0.138 |
| QuickSort Inserção m=10 | 5,972 | 5,454 | 0.003 |
| QuickSort Inserção m=100 | 17,189 | 16,321 | 0.011 |
| QuickSort vetor de inteiros | 4,907 | 4,859 | 0 |

| Tipo QuickSort (N=5000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 32,783 | 32,805 | 0.034 |
| QuickSort Mediana de 3 valores | 34,194 | 34,981 | 0.964 |
| QuickSort Mediana de 5 valores | 32,842 | 33,963 | 1.099 |
| QuickSort Inserção m=10 | 39,946 | 36,869 | 0.024 |
| QuickSort Inserção m=100 | 103,185 | 98,507 | 0.038 |
| QuickSort vetor de inteiros | 30,002 | 29,905 | 0.019 |

| Tipo QuickSort (N=10000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 71,530 | 71,016 | 0.068 |
| QuickSort Mediana de 3 valores | 78,130 | 79,977 | 2.403 |
| QuickSort Mediana de 5 valores | 83,146 | 84,489 | 2.3 |
| QuickSort Inserção m=10 | 90,870 | 85,772 | 0.089 |
| QuickSort Inserção m=100 | 226,423 | 217,544 | 0.102 |
| QuickSort vetor de inteiros | 71,030 | 70,950 | 0.05 |

| Tipo QuickSort (N=50000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 479,928 | 482,733 | 0.541 |
| QuickSort Mediana de 3 valores | 302,678 | 487,009 | 14.403 |
| QuickSort Mediana de 5 valores | 299,562 | 512,487 | 15.984 |
| QuickSort Inserção m=10 | 371,409 | 826,897 | 1.296 |
| QuickSort Inserção m=100 | 822,420 | 1,232,570 | 1.157 |
| QuickSort vetor de inteiros | 451,832 | 452,526 | 0.348 |

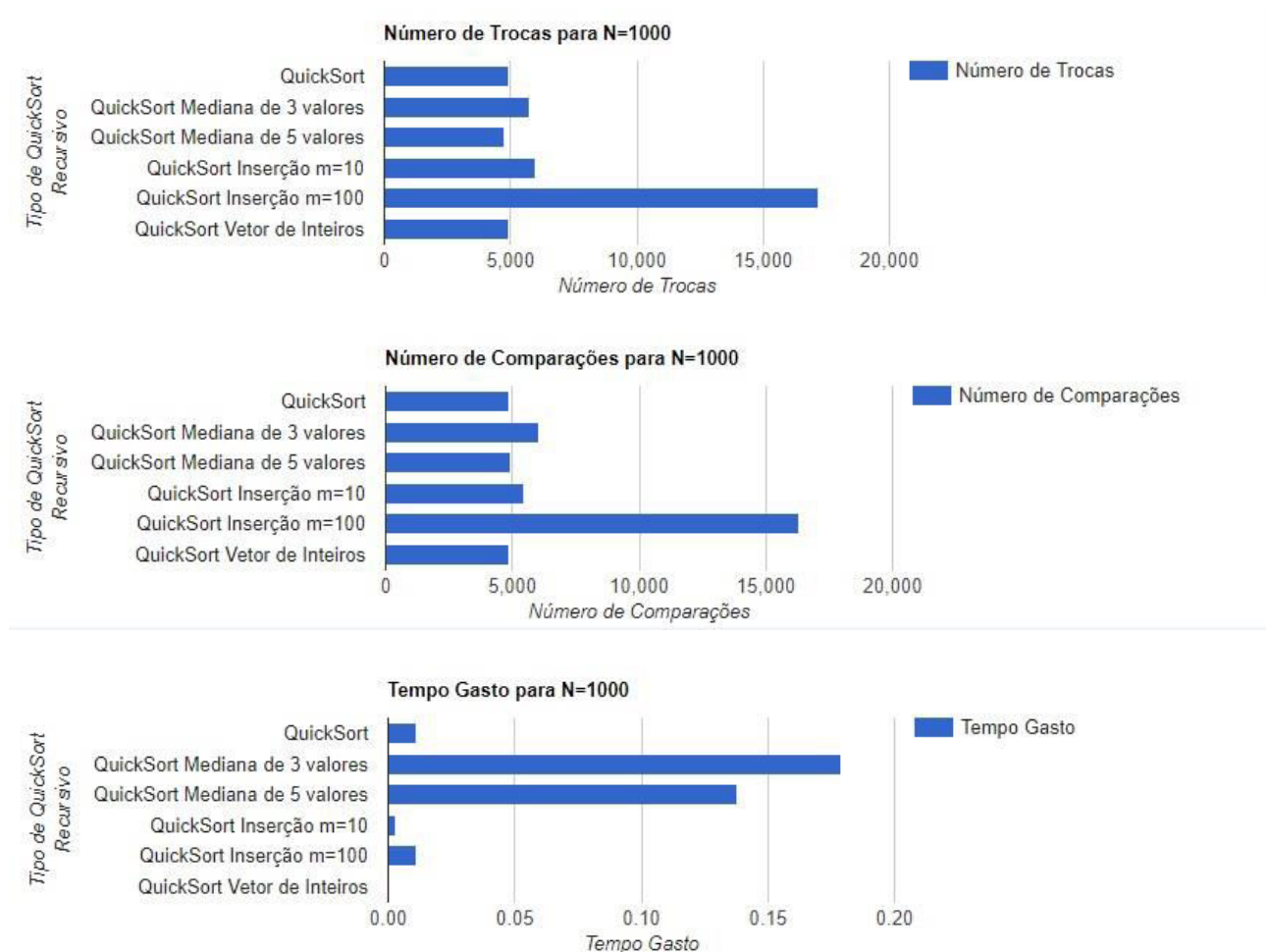
| Tipo QuickSort (N=100000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 4,292,790 | 7,982,920 | 37.438 |
| QuickSort Mediana de 3 valores | 377,150 | 1,049,330 | 33.068 |
| QuickSort Mediana de 5 valores | 332,572 | 1,020,020 | 31.032 |
| QuickSort Inserção m=10 | 410,862 | 887,900 | 0.967 |
| QuickSort Inserção m=100 | 859,399 | 1,196,560 | 0.808 |
| QuickSort vetor de inteiros | 1,036,320 | 1,037,590 | 0.72 |

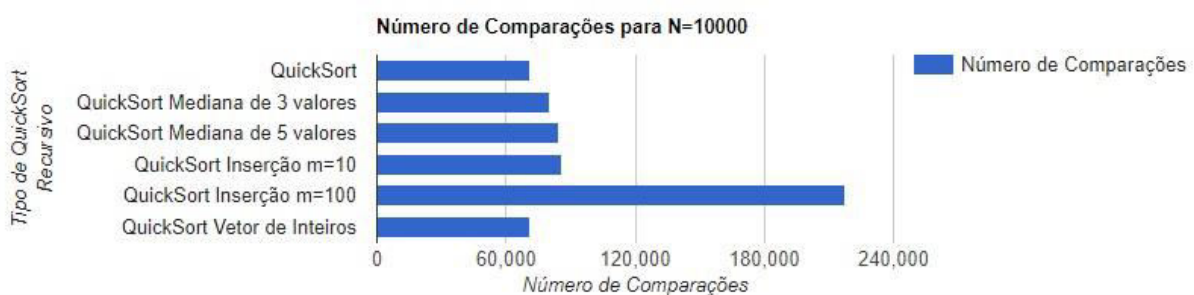
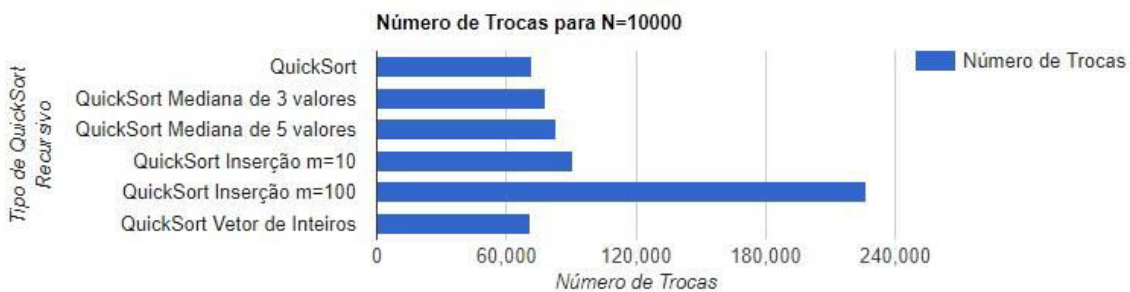
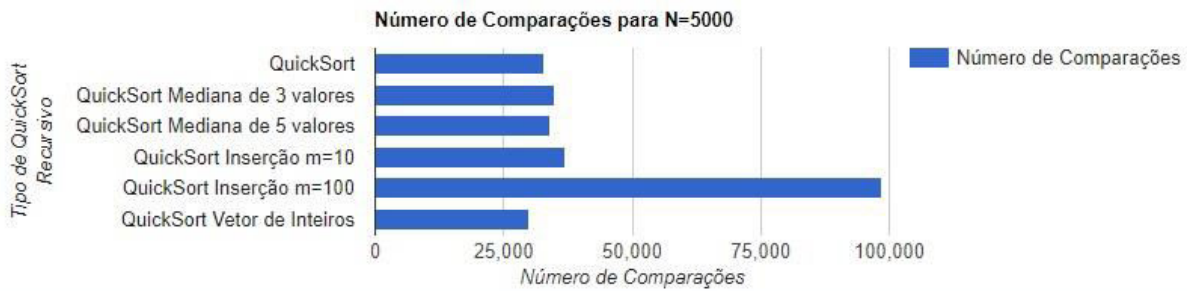
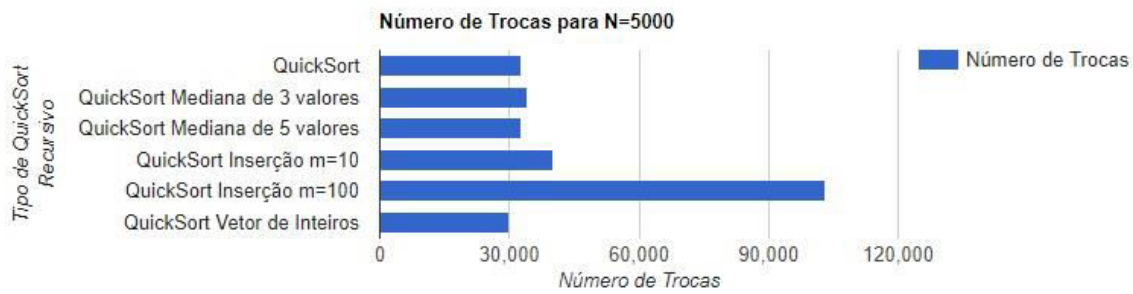
| Tipo QuickSort (N=500000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 6,019,220 | 6,123,370 | 10.679 |
| QuickSort Mediana de 3 valores | 937,853 | 5,713,780 | 192.996 |
| QuickSort Mediana de 5 valores | 649,043 | 5,682,040 | 156.361 |
| QuickSort Inserção m=10 | 820,779 | 4,834,950 | 5.161 |
| QuickSort Inserção m=100 | 1,266,160 | 4,379,130 | 3.84 |
| QuickSort vetor de inteiros | 5,570,400 | 5,582,330 | 4.953 |

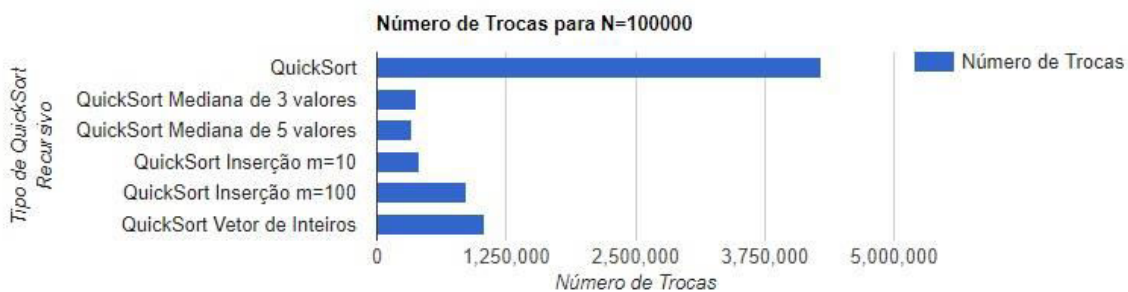
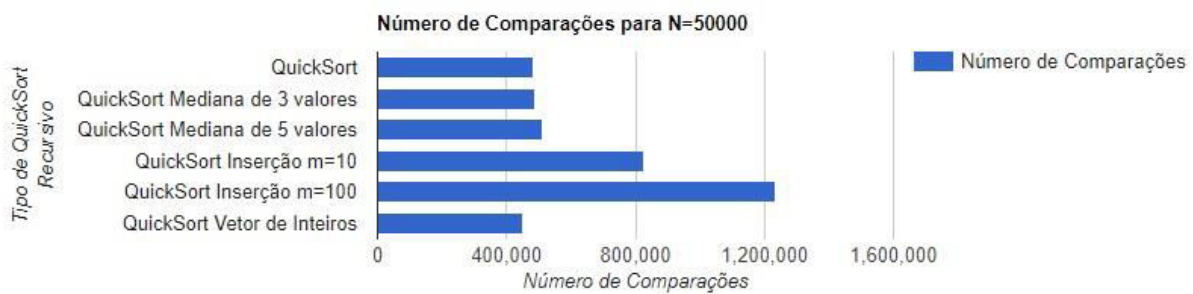
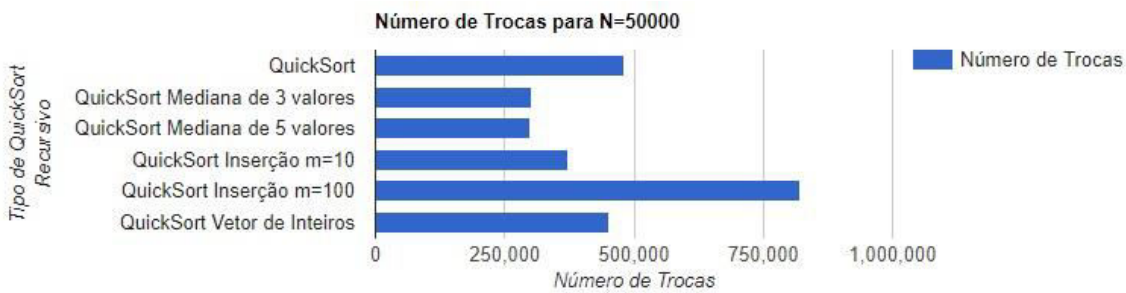
| Tipo QuickSort (N=1000000) | Número de trocas | Número de comparações | Tempo Gasto |
|--------------------------------|------------------|-----------------------|-------------|
| QuickSort | 12,172,300 | 14,091,100 | 30.161 |
| QuickSort Mediana de 3 valores | 1,146,250 | 13,598,200 | 361.7 |
| QuickSort Mediana de 5 valores | 1,036,800 | 13,012,800 | 364.732 |
| QuickSort Inserção m=10 | 1,321,860 | 10,343,100 | 12.586 |
| QuickSort Inserção m=100 | 1,771,370 | 8,947,220 | 7.743 |
| QuickSort vetor de inteiros | 12,899,100 | 12,925,200 | 11.039 |

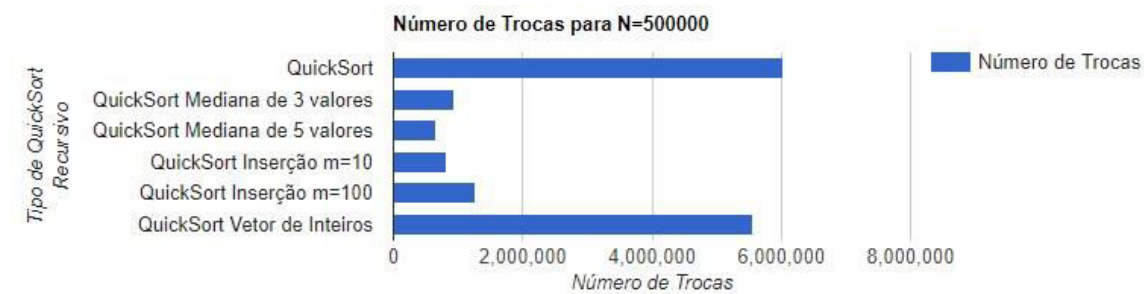
-Gráficos

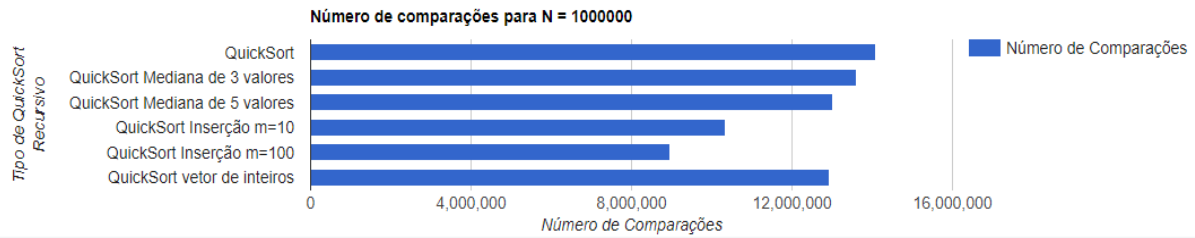
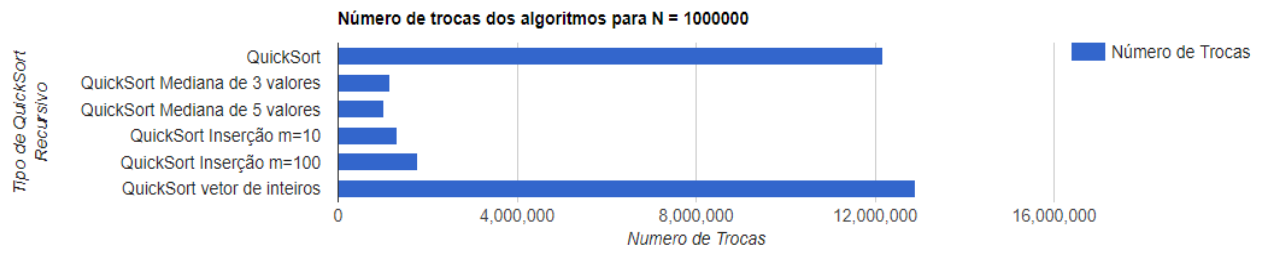
Os gráficos a seguir foram criados usando os dados retornados por cada tipo de QuickSort com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.











-Análise dos Algoritmos de Ordenação

-Tabelas

As tabelas a seguir foram criadas usando os dados retornados por cada algoritmo de ordenação com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.

De acordo com o desempenho o QuickSort escolhido para comparação foi o Algoritmo QuickSort Recursivo com Insercao com m=10.

Os algoritmos InsertionSort e BubbleSort apresentam alguns dados iguais a 0 a partir de N=100000, isso ocorreu devido ao tamanho do inteiro retornado durante os testes, eles apresentavam valores maiores do que o aceitável pelo tipo de variável int.

| Tipo QuickSort (N=1000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|-------------------------|------------------|-----------------------|-------------|
| QuickSort | 5,972 | 5,454 | 0.003 |
| MergeSort | 8,721 | 4,305 | 0.009 |
| InsertionSort | 251,880 | 250,881 | 0.011 |
| BubbleSort | 250,881 | 250,881 | 0.029 |
| HeapSort | 1,343 | 8,094 | 0.001 |

| Tipo Algoritmo (N=5000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|-------------------------|------------------|-----------------------|-------------|
| QuickSort | 39,946 | 36,869 | 0.024 |
| MergeSort | 55,213 | 27,074 | 0.043 |
| InsertionSort | 6,175,670 | 6,170,670 | 0.276 |
| BubbleSort | 6,170,670 | 6,170,670 | 0.797 |
| HeapSort | 6,771 | 52,126 | 0.005 |

| Tipo Algoritmo (N=10000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------|------------------|-----------------------|-------------|
| QuickSort | 90,870 | 85,772 | 0.089 |
| MergeSort | 120,429 | 58,994 | 0.111 |
| InsertionSort | 24,749,700 | 24,739,700 | 1.079 |
| BubbleSort | 24,739,700 | 24,739,700 | 3.102 |
| HeapSort | 13,454 | 114,559 | 0.013 |

| Tipo Algoritmo (N=50000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|--------------------------|------------------|-----------------------|-------------|
| QuickSort | 371,409 | 826,897 | 1.296 |
| MergeSort | 658,572 | 360,490 | 0.656 |
| InsertionSort | 677,285,000 | 677,235,000 | 37.576 |
| BubbleSort | 677,235,000 | 677,235,000 | 128.194 |
| HeapSort | 73,178 | 677,271 | 0.1 |

| Tipo Algoritmo (N=100000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|---------------------------|------------------|-----------------------|-------------|
| QuickSort | 410,862 | 887,900 | 0.967 |
| MergeSort | 1,238,170 | 635,350 | 1.521 |
| InsertionSort | 0 | 0 | 230.356 |
| BubbleSort | 0 | 0 | 571.805 |
| HeapSort | 145,462 | 1,464,440 | 0.2 |

| Tipo Algoritmo (N=500000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|---------------------------|------------------|-----------------------|-------------|
| QuickSort | 820,779 | 4,834,950 | 5.161 |
| MergeSort | 6,991,180 | 4,002,860 | 8.968 |
| InsertionSort | 0 | 0 | 0 |
| BubbleSort | 0 | 0 | 0 |
| HeapSort | 738,385 | 8,417,840 | 1.226 |

| Tipo Algoritmo (N=1000000) | Numero de trocas | Numero de comparacoes | Tempo Gasto |
|----------------------------|------------------|-----------------------|-------------|
| QuickSort | 1,321,860 | 10,343,100 | 12.586 |
| MergeSort | 13,126,900 | 5,302,360 | 18.151 |
| InsertionSort | 0 | 0 | 0 |
| BubbleSort | 0 | 0 | 0 |
| HeapSort | 1,452,430 | 18,076,600 | 2.551 |

-Gráficos

Os gráficos a seguir foram criados usando os dados retornados por cada algoritmo de ordenação com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.

De acordo com o desempenho o QuickSort escolhido para comparação foi o Algoritmo QuickSort Recursivo com Insercao com m=10.

Os algoritmos InsertionSort e BubbleSort apresentam alguns dados iguais a 0 a partir de N=100000, isso ocorreu devido ao tamanho do inteiro retornado durante os testes, eles apresentavam valores maiores do que o aceitável pelo tipo de variável int.



