

Nome: Ruan Pablo Vitkoski de Souza

22/09/2023

**GEX609 - PESQUISA E ORDENAÇÃO DE DADOS - T01 (2023.2)**

Nome da Tarefa:

Trabalho Prático 1 (TP1)

Descrição:

Usar o templateT1.c disponível no git, para comparar 4 métodos de ordenação: Bubble Sort, Insertion Sort, Selection Sort e o Quick Sort em relação ao número de trocas e o tempo de execução.

Os códigos do Bubble, Insertion e o Selection poderão ser reaproveitados dos trabalhos anteriores. Já o código do QuickSort, terá de ser implementado e explicado.

A comparação dos métodos irá resultar em dois gráficos variando o tamanho da entrada entre 1000, 5000, 10000, 50000, 100000 e 1000000. O primeiro gráfico irá reportar o número de trocas realizadas por cada algoritmo em cada entrada. Já o segundo, irá reportar o tempo gasto por cada método para o ordenamento. É importante que o tempo seja a média de 3 execuções.

O trabalho poderá ser feito em duplas ou individual e será apresentado individualmente.

**Programa:**

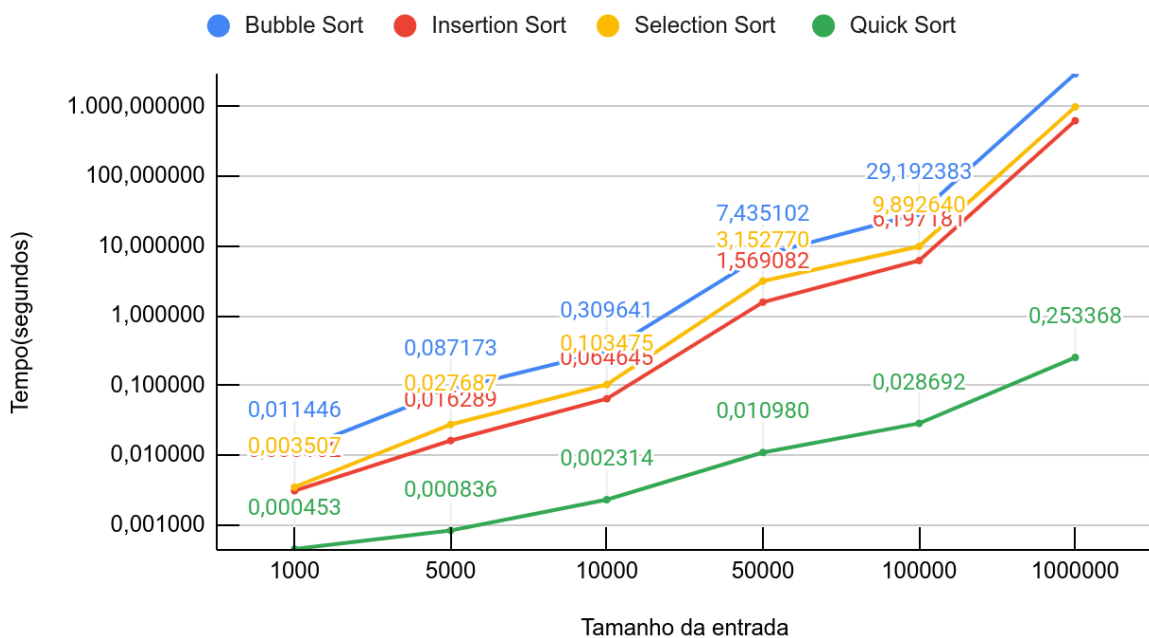
[https://github.com/Ruan2PVS9/uffs\\_cc\\_repository/blob/main/POD/Trabalho\\_1/trabalho\\_p1\\_ruan.c](https://github.com/Ruan2PVS9/uffs_cc_repository/blob/main/POD/Trabalho_1/trabalho_p1_ruan.c)

### Coleta de dados:

Com a coleta dos dados das execuções foi pego a seguintes médias:

tempo/tamanho				
	Bubble Sort	Insertion Sort	Selection Sort	Quick Sort
1000	0,011446	0,003102	0,003507	0,000453
5000	0,087173	0,016289	0,027687	0,000836
10000	0,309641	0,064645	0,103475	0,002314
50000	7,435102	1,569082	3,152770	0,010980
100000	29,192383	6,197181	9,892640	0,028692
1000000	2950,253339	627,144585	991,628726	0,253368

Tempo / tamanho de entrada

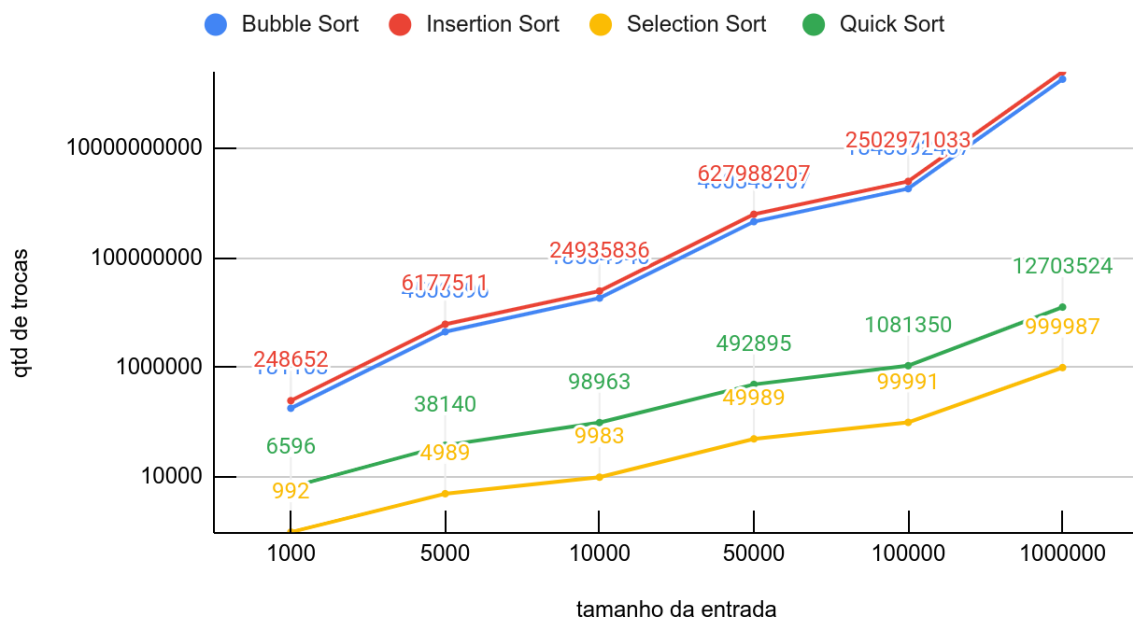


na consideração de tempo por tamanho da entrada de elementos se percebe o grande benefício  $O(N \cdot \log N)$  sendo que o método quick sort levando muito menos tempo, enquanto os demais métodos se demonstraram um tanto parecido em comparação a eficiência do método quick.

ja na quantidade de trocas:

Trocas/tamanho				
	Bubble Sort	Insertion Sort	Selection Sort	Quick Sort
1000	181103	248652	992	6596
5000	4508596	6177511	4989	38140
10000	18554940	24935836	9983	98963
50000	460845107	627988207	49989	492895
100000	1843392467	2502971033	99991	1081350
1000000	183791386460	249838703583	999987	12703524

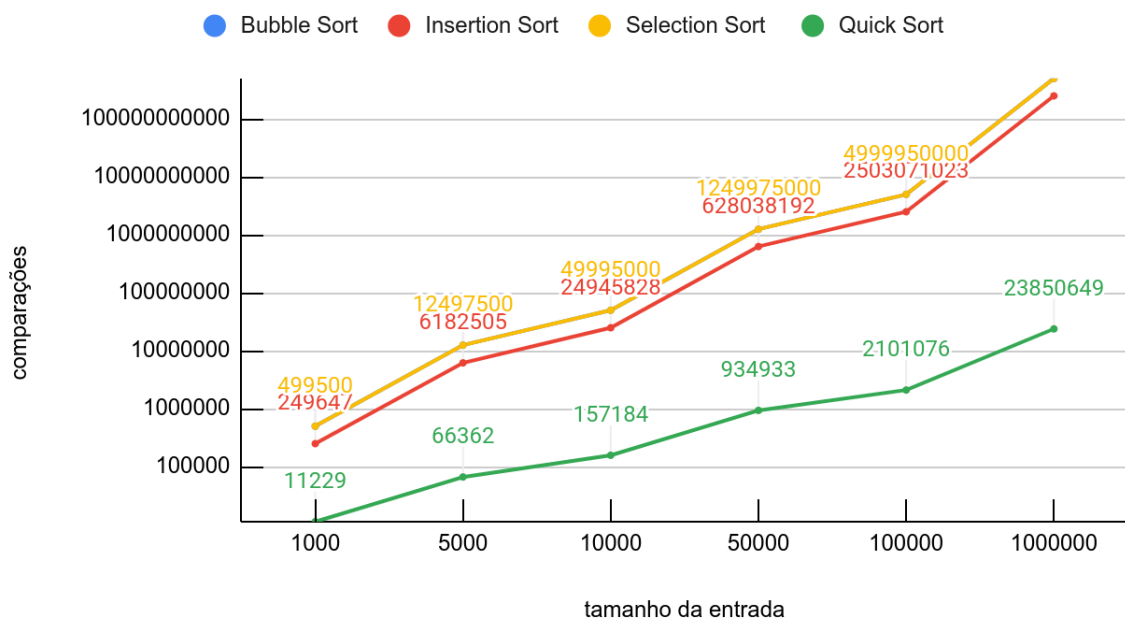
trocas / tamanho de entrada



O método selection Sort se mostrou o que faz um número menor de trocas, efetuando menos trocas até mesmo que o quickSort.

comparações/tamanho				
	Bubble Sort	Insertion Sort	Selection Sort	Quick Sort
1000	499500	249647	499500	11229
5000	12497500	6182505	12497500	66362
10000	49995000	24945828	49995000	157184
50000	1249975000	628038192	1249975000	934933
100000	4999950000	2503071023	4999950000	2101076
1000000	499999500000	249839703567	499999500000	23850649

comparações/tamanho



Mas a quantidade de comparações feitas é muito parecida, sendo que Selection Sort fez a mesma quantidade que o Bubble Sort.