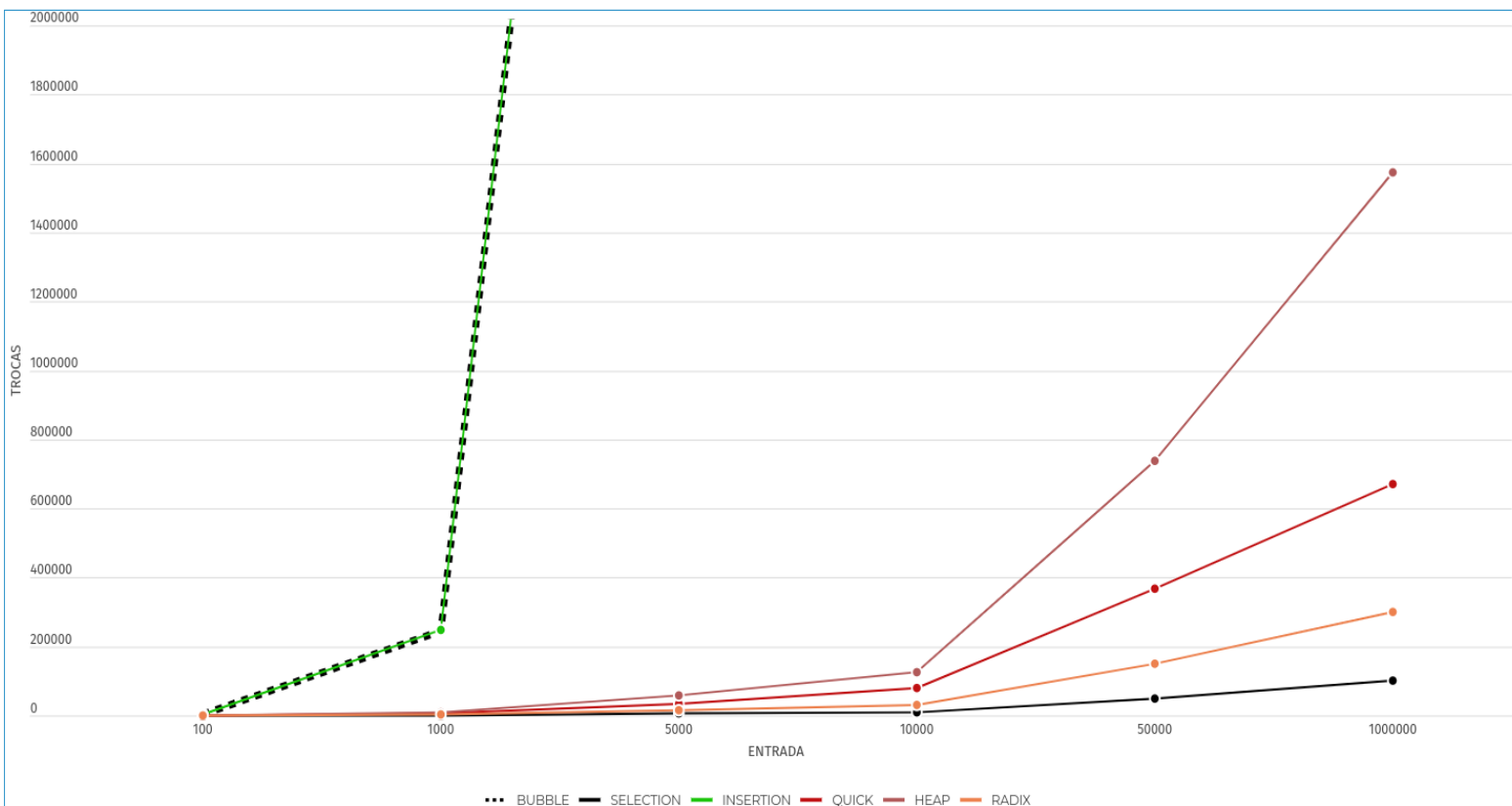


## Número de trocas



[08]

Este é o comportamento das curvas dos métodos de ordenação para vetores aleatórios de 100, 1000, 5000, 10000, 50000 e 100000 elementos. O vetor randômico é preenchido com números de 0 a 1000.

Na imagem acima, o número de trocas(eixo vertical) foi limitado em 2 milhões de trocas para melhor visualização. Pode-se perceber que tanto o bubble quanto o insertion apresentam, com uma entrada “baixa” de 5000 elementos, um aumento muito grande no número de trocas em comparação com uma entrada de 1000 elementos. Isso acontece pois ambos são algoritmos quadráticos, ou seja, possuem um 2 laços for, um dentro do outro. Com isso, devido a linha 163, da função bubble, e linha 178, da função insertion, onde há trocas de elementos dentro de 2 laços for, o número de trocas desses dois algoritmos é muito maior quando comparado aos outros deste relatório.

Por outro lado, o selection é o algoritmo que apresenta o menor número de trocas, menor até que o número de elementos do vetor. Isso acontece nas linhas 201, 203 e 203, onde a troca acontece dentro de um laço for apenas, e não de dois.

As trocas do quick sort acontecem na linha 215, onde está trocando de posição os valores menores que o pivô, e na linha 219, onde está inserindo o pivô na sua posição correta.

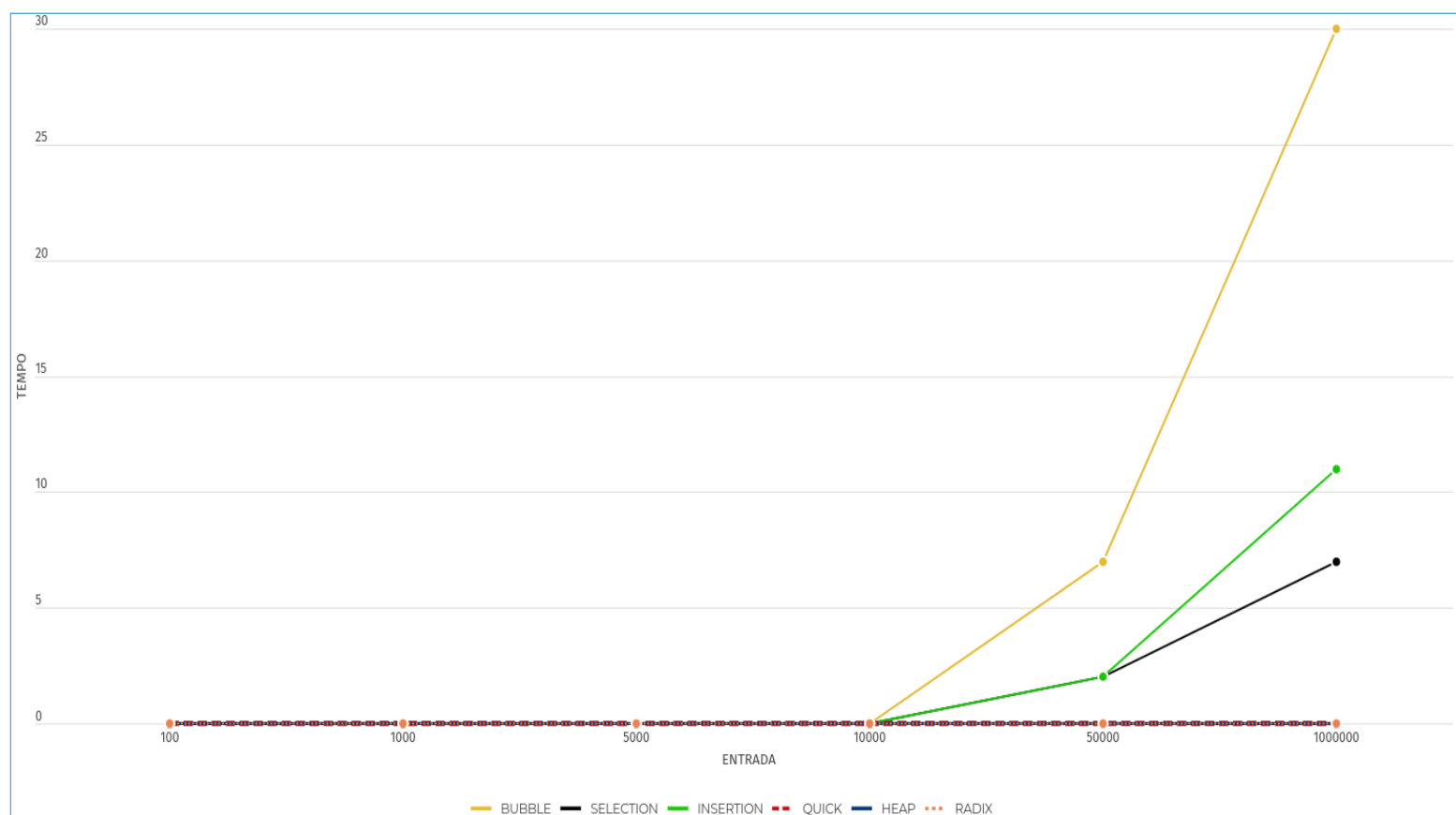
As trocas do heap sort ocorrem na linha 244, onde troca os elementos caso o nó filho seja menor que o nó pai, e na linha 256, onde troca a raiz do heap máximo com o último elemento não ordenado do vetor.

Já no radix sort, o único momento em que há trocas no vetor original é no momento da cópia do vetor auxiliar. Isso acontece na linha 295.

Abaixo temos uma tabela com os dados absolutos de número de trocas para todos os métodos de ordenação em cada tamanho de entrada.

ENTRADA	BUBBLE	SELECTION	INSERTION	QUICK	HEAP	RADIX
100	2387	96	2486	359	587	300
1000	245765	990	246764	5842	9091	3000
5000	6257767	4985	6262766	33351	57085	15000
10000	24765785	9980	24775784	78152	124202	30000
50000	623986515	49945	624019847	366537	736983	150000
1000000	2497397677	99896	2497497676	669706	1573976	300000

### Tempo de execução



O gráfico de tempo mostra que até a entrada de 10000 elementos, todos os algoritmos operam abaixo de 1 segundo, mas a partir disso, vemos que o bubble e o insertion têm um grande aumento no tempo de execução, justamente por serem algoritmos quadráticos. O selection também apresentou um considerável aumento no tempo de execução, devido ao seu alto número

de comparações. Os demais algoritmos (quick, heap e radix) apresentaram tempo de execução abaixo de um segundo para todas as entradas, tanto que no gráfico as linhas estão praticamente sobrepostas.

Abaixo vemos uma tabela com os dados absolutos de tempo de execução

ENTRADA	BUBBLE	SELECTION	INSERTION	QUICK	HEAP	RADIX
100	0,0000620	0,0000283	0,0000297	0,00001433	0,0000250	0,0000170
1000	0,0040130	0,0017623	0,0021517	0,00015467	0,0003477	0,0001813
5000	0,0541260	0,0178800	0,0284660	0,00041500	0,0008933	0,0004310
10000	0,2320813	0,0730897	0,1149573	0,00108533	0,0018790	0,0010680
50000	7,3159273	2,0251250	2,8153300	0,00617933	0,0122247	0,0041860
1000000	30,1203123	7,8337960	11,2694570	0,01535467	0,0250630	0,0085883

Com base na tabela, vemos que o radix sort foi o que apresentou o menor tempo de execução.