

**Ambiente de teste:**

**Hardware:** Notebook Lenovo Yoga 510, I5 6200U, 4Gb RAM.

**Sistema Operacional:** Linux Elementary OS.

**Elementos:** Números aleatórios entre 0 e o tamanho do vetor \* 2.

BubbleSort:

Elementos	Tempo de execução
100	0.003858699 segundos
1000	0.029141444 segundos
10000	0.246969468 segundos
20000	0.756836882 segundos
30000	1.684148184 segundos
40000	3.06228993 segundos
50000	4.784795661 segundos
60000	7.491254665 segundos
70000	9.531259817 segundos
80000	12.407673848 segundos

O tempo estourou com 80 mil elementos.

**Comparação com a complexidade teórica do algoritmo:** A complexidade do algoritmo é de ordem quadrática, é possível constatar na prática com os tempos de execução das ordenações, os tempos se aproximam dessa complexidade teórica, sendo levado em conta, a máquina onde está sendo executado o código, os elementos que compõem o vetor, etc.

QuickSort:

Elementos	Tempo de execução
100	0.002401537 segundos*
1000	0.000865077 segundos
10000	0.002070673 segundos
20000	0.001995293 segundos*
30000	0.004128703 segundos
40000	0.004183291 segundos
50000	0.005334868 segundos
60000	0.006523608 segundos
70000	0.026951768 segundos

Testando, foi constatado que o tempo de 10 segundos estoura em aproximadamente 68 milhões e 500 mil elementos.

**Comparação com a complexidade teórica do algoritmo:** Em alguns pontos, como no 100 elementos, e no 20k elementos, houve uma divergência no tempo de execução, em comparação aos outros tempos, em relação aos outros tempos é seguido um padrão de tempo, possivelmente, essa divergência pode ter sido causada por alguma circunstância da linguagem, ou de que era um pior caso, tendo em vista que o pior caso do Quicksort é  $O(n^2)$ , com isso os tempos sofrem influência de acordo com a disposição dos elementos, tendendo ao melhor caso e caso médio.

**Fibonacci:**

No ambiente de teste, o algoritmo iniciando com  $N=3$ , foi capaz de realizar 47 execuções, até estourar o tempo de 30s. Ele estourou o tempo no número <-298632863>, com o tempo de 42 segundos.