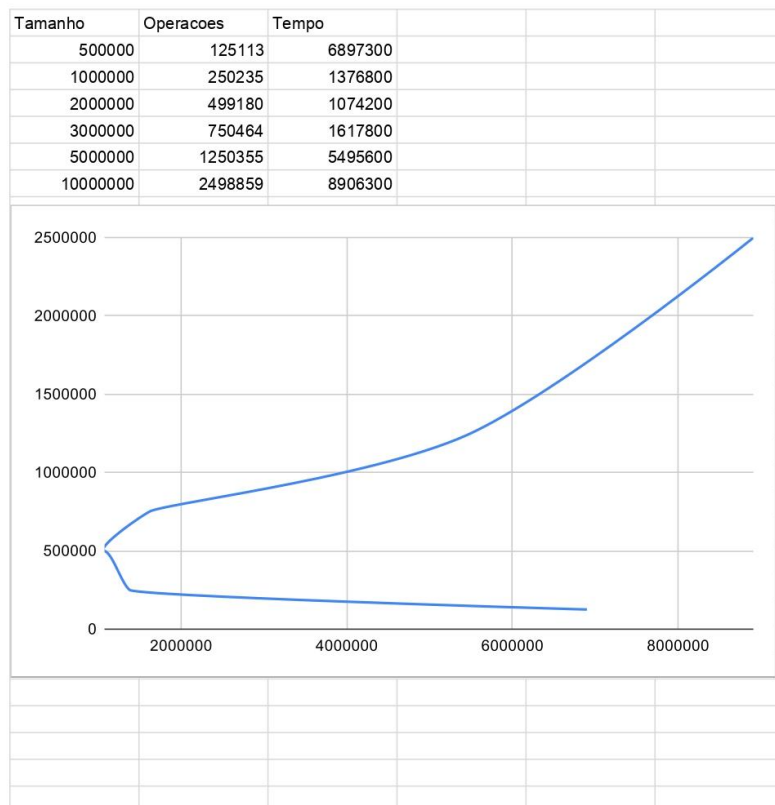


# Oficina 1

Código 1:

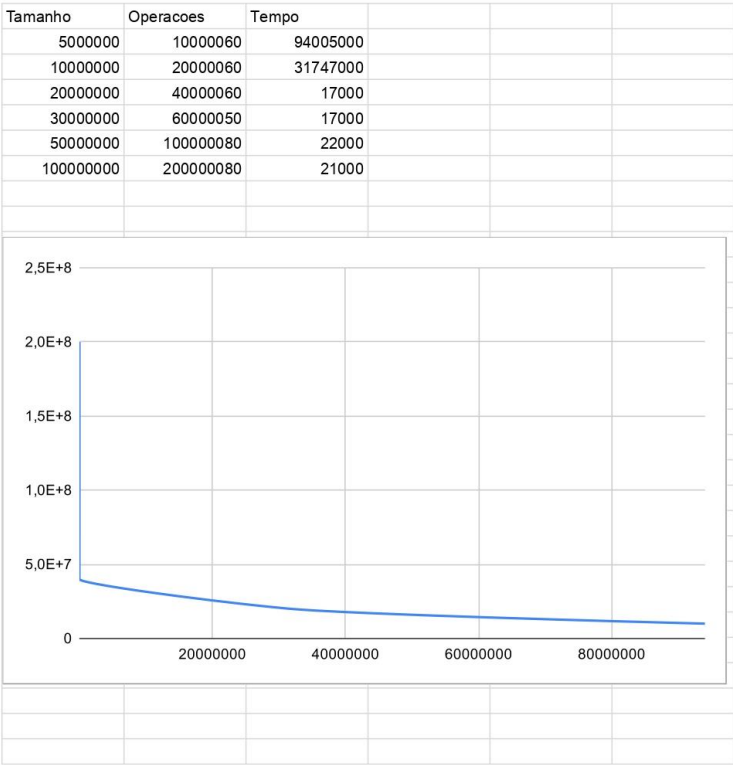
Os dados indicam que o tempo de execução do Algoritmo 1 cresce de forma não linear com o aumento do tamanho do vetor, o que pode sugerir um crescimento quadrático. Podemos observar um aumento significativo quando o tamanho do vetor cresce, o que é esperado em algoritmos mais complexos como o Selection Sort.



Código 2:

O Algoritmo 2 apresenta uma redução considerável no tempo de execução quando comparado com o Algoritmo 1. A maior parte do tempo de execução é concentrada nos primeiros tamanhos de vetor, como 500.000 e 1.000.000, mas o tempo se estabiliza e diminui a partir de 2.000.000.

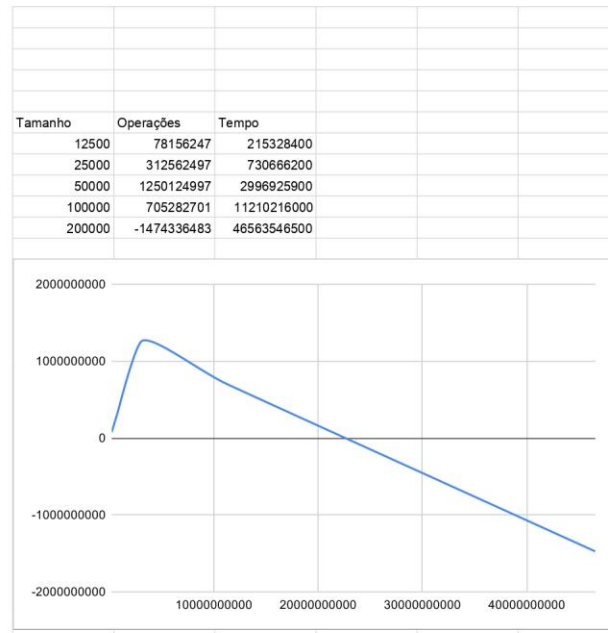
É provável que a complexidade do Algoritmo 2 seja  $O(n \log n)$ , o que é mais eficiente para grandes inputs.



Código 3:

Este comportamento sugere que o algoritmo tem uma complexidade quadrática  $O(n^2)$  ou superior, fazendo com que seu desempenho degrade rapidamente à medida que o tamanho do vetor cresce.

Erro no  $N=200000$  porque o número de operações superou o limite máximo do tipo de dado int em Java.



Código 4:

O número de operações desse algoritmo cresce exponencialmente, o tempo de execução também cresce exponencialmente, o pior algoritmo para grandes inputs

