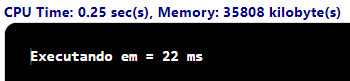
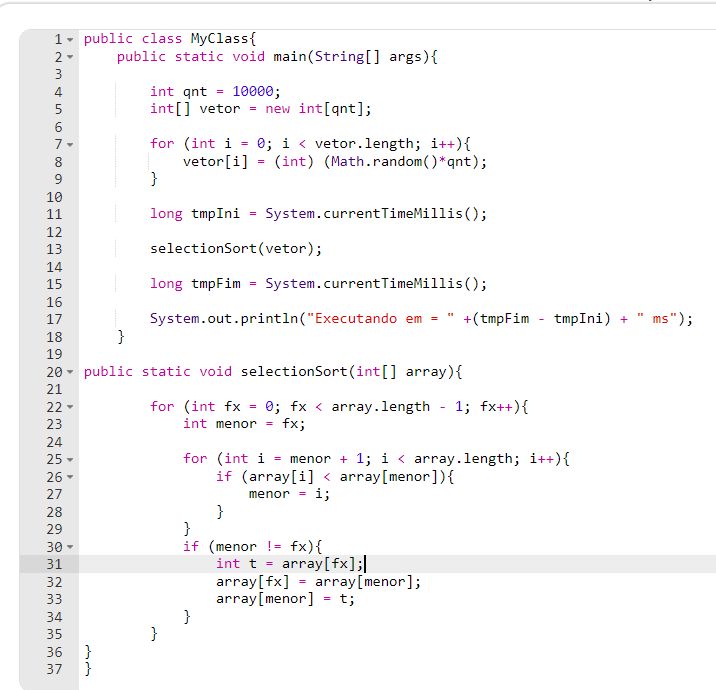
**Matheus Henrique de Oliveira Querido**

**Comparar o Selection, Insertion e Bubble**

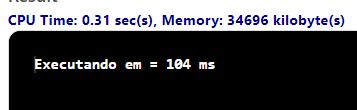
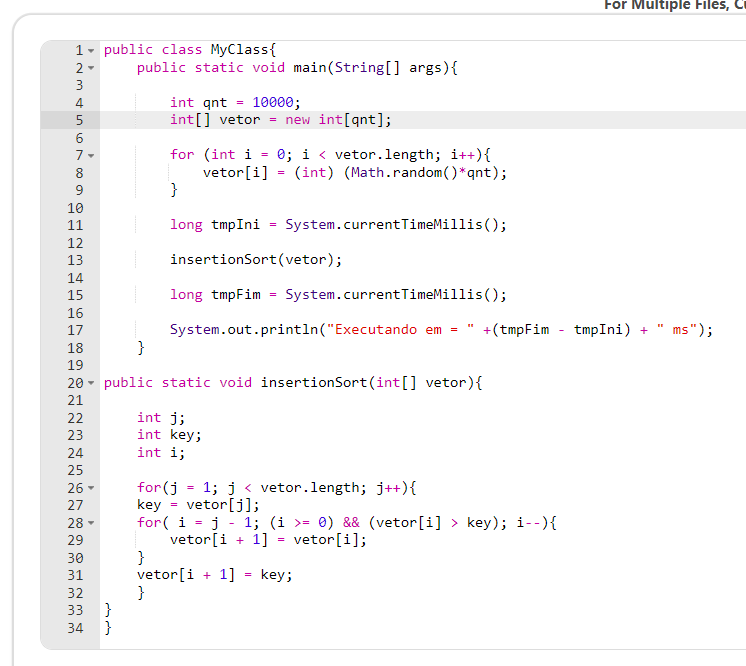
**Selection Sort**

Esse algoritmo funciona da seguinte maneira: dois números são selecionados onde o menor valor é colocado na primeira posição e o maior na segunda, é o código com o menor tempo de resposta.



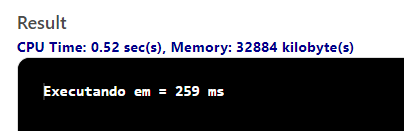
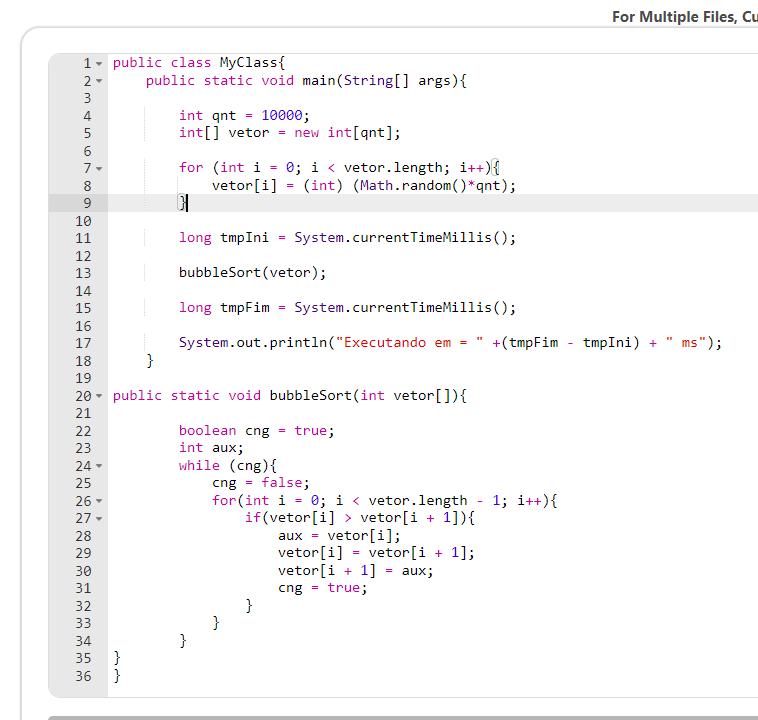
**Insertion Sort**

Nesse caso, é construído uma matriz com uma inserção de cada vez, é um algoritmo de ordenação bem eficiente para pequenas entradas/listas, é um pouco mais lento que o Selection Sort.



**Bubble Sort**

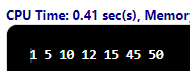
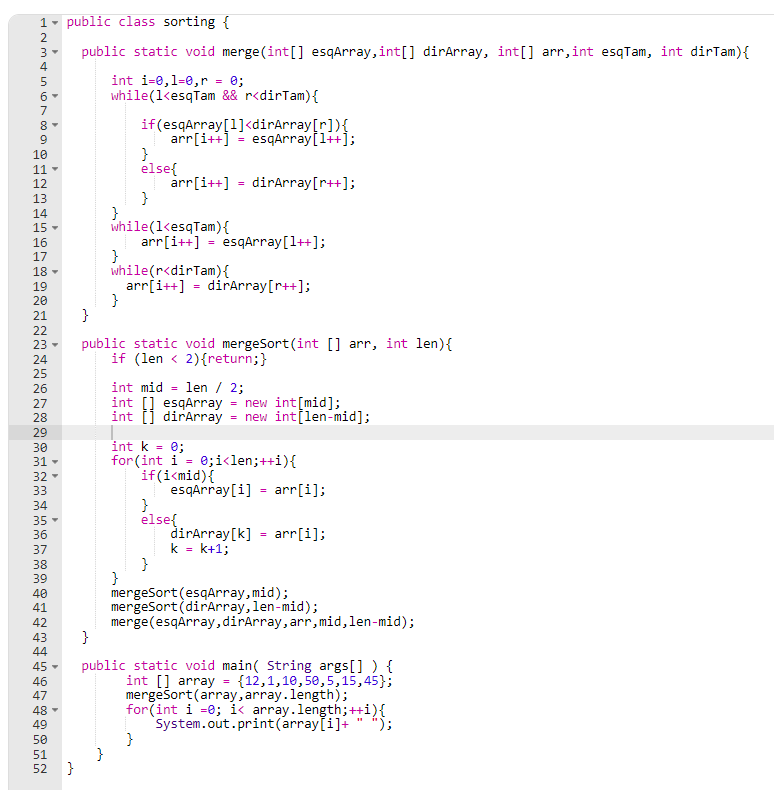
Nesse caso, o algoritmo é o mais complexo em relação aos demais. Ele realiza uma comparação com o elemento anterior ao posterior. Esse algoritmo é o que levou mais tempo entre todos esses.



**Comparar o Merge, Quick e Arrays.sort().**

**Merge Sort**

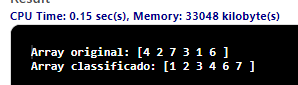
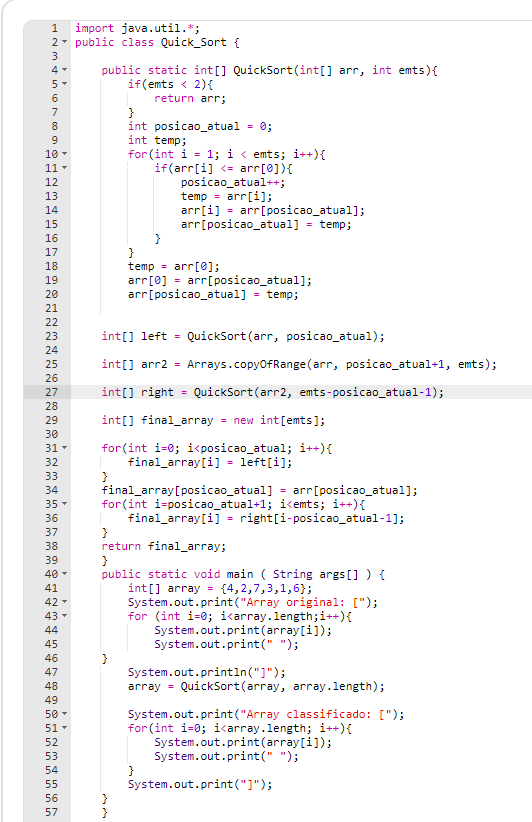
Combina dois arrays ordenados em um outro também ordenado, após a execução, é possível ver que é um pouco mais lento que o Quick Sort.



****

**Quick Sort**

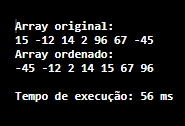
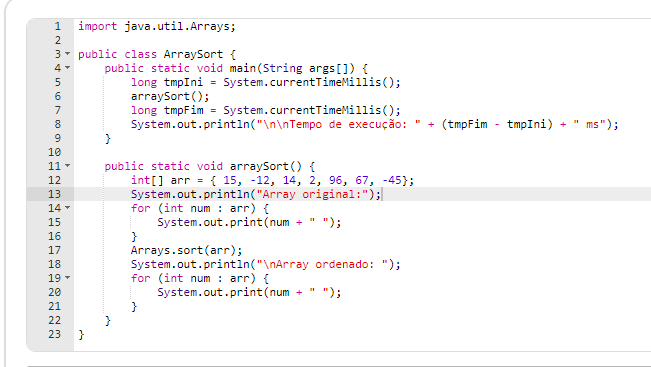
Adota a estratégia divisão e conquista. Basicamente é adotado um valor aleatório dentro do array que é chamado de pivô e posiciona todos elementos de maneira que os menores ou iguais se posicionem a esquerda e os maiores á direita, é mais rápido que o Array Sort.





**Array Sort**

O seu tempo de resposta pode variar muito, pois é feito uma comparação um a um até encontrar a resposta.



**Comparar busca simples, binária e Arrays.binarySearch().**

O método de busca simples percorre um array para realizar a comparação das chaves, para verificar se o valor da chave é o mesmo naquele momento. O método deve estar inserido dentro de um array.

Já o método de busca binária tem como foco também buscar não só os elementos dentro do array, como também os de fora. Ele nos trás dois tipos de mensagem, que diz se existe ou não o item e também irá se a chave é maior ou menor que a metade do seu valor.