

# Relatório Métodos de Ordenação C#

## Introdução:

**Bubble Sort**: É o mais simples que tem, percorre todo o vetor trocando o menos pelo maior. Mais lento.

**Selection Sort**: Verifica todos os números, seleciona o menor e coloca em primeiro. Melhor para listas/vetores menores.

**Insertion Sort**: Seleciona o menor número e o coloca antes do número que o precede (ex. 5 antes de 6). Não é muito eficaz em listas muito grandes, pois aumenta a quantidade de trocas que se faz, colocando todos os números atrás do que o precede até chegar a posição correta.

**Quick Sort**: Seleciona um elemento denominado pivô e organiza a lista de forma que os números anteriores a ele são menores e os posteriores são maiores, repete até ordenar. Um dos métodos mais rápidos e eficaz, pois ordena os dois lados da lista, tendo 1 número pivô como base.

**Merge Sort**: Divide o vetor em sub-vetores com no máximo 2 números cada e organiza os sub-vetores, colocando o menor antes do maior, ao organizar os sub-vetores, junta eles e organiza de novo até ficar ordenado. É o mais rápido, pois afeta vários números por vez.

## Implementação:

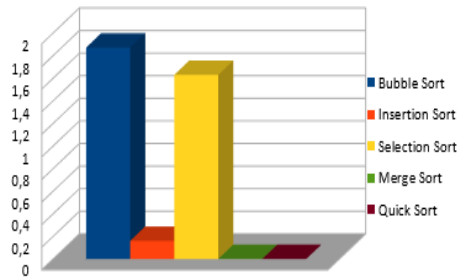
A principal implementação que encontrei foi usar o **for** dentro do **for** com um **if** dentro do segundo **for** para controlar quando ele é acionado e quando não é. No **Merge Sort**

em especial, foi usado **while** e **else** para controlar os sub-vetores e, trocar suas posições usando o **temp** e o método de troca de vetores.

Analise de complexidade:

Bubble Sort	Insertion Sort	Selection Sort	Merge Sort	Quick Sort
1 Minutos, 55 Segundo e 285 Milisegundos	0 Minutos, 9 Segundo e 904 Milisegundos	1 Minutos, 42 Segundo e 132 Milisegundos	0 Minutos, 0 Segundo e 45 Milisegundos	0 Minutos, 0 Segundo e 20 Milisegundos
1 Minutos, 55 Segundo e 833 Milisegundos	0 Minutos, 9 Segundo e 422 Milisegundos	1 Minutos, 41 Segundo e 695 Milisegundos	0 Minutos, 0 Segundo e 45 Milisegundos	0 Minutos, 0 Segundo e 17 Milisegundos
1 Minutos, 57 Segundo e 795 Milisegundos	0 Minutos, 9 Segundo e 894 Milisegundos	1 Minutos, 41 Segundo e 160 Milisegundos	0 Minutos, 0 Segundo e 41 Milisegundos	0 Minutos, 0 Segundo e 19 Milisegundos
1,867338837	0,165128418	1,623760217	0,000694118	0,00041622
1,87226088	0,155898363	1,6274807	0,0007276	0,000319988
1,875455692	0,16267159	1,637311512	0,00065015	0,00031085
1,871685136	0,161232791	1,629517476	0,000690623	0,000349019

Tempo de Execução em minutos



Conclusão:

Trabalho muito complexo, tive que pesquisar bastante para conseguir fazer, a principal dificuldade encontrada foi não conhecer muito bem os métodos de ordenação. Obtive os seguintes resultados com cada um dos 3 vetores:

Bublle Sort:

1 Minutos, 55 Segundo e 285 Milisegundos
1 Minutos, 55 Segundo e 833 Milisegundos
1 Minutos, 57 Segundo e 795 Milisegundos

### **Selection Sort:**

0 Minutos, 9 Segundo e 904 Milisegundos
0 Minutos, 9 Segundo e 422 Milisegundos
0 Minutos, 9 Segundo e 894 Milisegundos

### **Insertion Sort:**

1 Minutos, 42 Segundo e 132 Milisegundos
1 Minutos, 41 Segundo e 695 Milisegundos
1 Minutos, 41 Segundo e 160 Milisegundos

### **Quick Sort:**

0 Minutos, 0 Segundo e 20 Milisegundos
0 Minutos, 0 Segundo e 17 Milisegundos
0 Minutos, 0 Segundo e 19 Milisegundos

### **Marge Sort:**

0 Minutos, 0 Segundo e 45 Milisegundos
0 Minutos, 0 Segundo e 45 Milisegundos
0 Minutos, 0 Segundo e 41 Milisegundos