

TDE 03 - Ordenação RA 04

Victor Silva Camargo, Vinícius Silva Camargo

¹Pontifícia Universitária Católica do Paraná

1. Bubble Sort

O método de ordenação Bubble sort usa a lógica de percorrer o vetor comparando o elemento em que ele está com o próximo, caso esteja desordenado, ele troca as posições do elemento atual com o próximo, e repete esse processo em uma quantidade fixa de vezes, a quantidade é sempre o tamanho do vetor - 1, esse método demonstra ser um dos mais lentos métodos de ordenação, devido a ele ter que percorrer todo o vetor, o tamanho todo dele - 1, independentemente se todas as trocas já foram feitas, pode ser ajustado com algum tipo de validador, como um booleano, por exemplo.

2. Insert Sort

O método de ordenação Insert Sort pega um vetor desordenado e faz a ordenação movendo os elementos não ordenados para a posição correta, pegando um elemento e fazendo comparação com os outros até chegar na sua devida posição.

3. Merge Sort

O método de ordenação Merge Sort usa a lógica de separar o vetor pela metade até que sobrem dois elementos. Alguns casos um elemento fica sozinho devido ao tamanho do vetor ser ímpar. Depois que divide, ele junta novamente por pares, só que dessa vez ordenando, até chegar em um vetor final ordenado. Esse método demonstrou ser mais eficiente em casos que o vetor está no pior caso de ordenação, além de que no geral, ele realiza menos trocas que os outros métodos de ordenação.

4. Shell Sort

O Shell Sort funciona de maneira parecida com o Insert Sort. O Shell Sort percorre o vetor e faz a divisão dele em sub-listas. Podemos determinar quantas sub-listas, mas no caso deste trabalho ele é definido com um grande incremento (chamado "span") e, em cada iteração, divide as sublistas internamente para classificar essas sublistas com base no "span". O "span" é determinado pelos valores em vetor[incr], que são recuperados a cada iteração.

```
Média de tempo de ordenação para 50 elementos: 33160 ns
Número médio de trocas: 581
Número médio de iterações: 49

Média de tempo de ordenação para 500 elementos: 884080 ns
Número médio de trocas: 63131
Número médio de iterações: 499

Média de tempo de ordenação para 1000 elementos: 1454980 ns
Número médio de trocas: 250553
Número médio de iterações: 999
|
Média de tempo de ordenação para 5000 elementos: 15488080 ns
Número médio de trocas: 6242272
Número médio de iterações: 4999

Média de tempo de ordenação para 10000 elementos: 95428380 ns
Número médio de trocas: 25010625
Número médio de iterações: 9999
```

Figure 1. Resultados da avaliação Bubble Sort

```
Média de tempo de ordenação para 50 elementos: 21580 ns
Média de trocas para 50 elementos: 660
Média de iterações para 50 elementos: 49

Média de tempo de ordenação para 500 elementos: 484900 ns
Média de trocas para 500 elementos: 62379
Média de iterações para 500 elementos: 499

Média de tempo de ordenação para 1000 elementos: 551660 ns
Média de trocas para 1000 elementos: 248702
Média de iterações para 1000 elementos: 999

Média de tempo de ordenação para 5000 elementos: 1674180 ns
Média de trocas para 5000 elementos: 6218849
Média de iterações para 5000 elementos: 4999

Média de tempo de ordenação para 10000 elementos: 7600020 ns
Média de trocas para 10000 elementos: 24959144
Média de iterações para 10000 elementos: 9999
```

Figure 2. Resultados da avaliação Insert Sort

```
Média de tempo de ordenação para 50 elementos: 36620 ns
Número médio de trocas: 103
Número médio de iterações: 49

Média de tempo de ordenação para 500 elementos: 100240 ns
Número médio de trocas: 1908
Número médio de iterações: 499

Média de tempo de ordenação para 1000 elementos: 158040 ns
Número médio de trocas: 4314
Número médio de iterações: 999

Média de tempo de ordenação para 5000 elementos: 1484840 ns
Número médio de trocas: 26649
Número médio de iterações: 4999

Média de tempo de ordenação para 10000 elementos: 2498320 ns
Número médio de trocas: 58262
Número médio de iterações: 9999
```

Figure 3. Resultados da avaliação Merge Sort

```
Média de tempo de ordenação para 50 elementos: 1740 ns
Média de trocas para 50 elementos: 3
Média de iterações para 50 elementos: 49

Média de tempo de ordenação para 500 elementos: 200840 ns
Média de trocas para 500 elementos: 5250
Média de iterações para 500 elementos: 499

Média de tempo de ordenação para 1000 elementos: 480240 ns
Média de trocas para 1000 elementos: 7392
Média de iterações para 1000 elementos: 999

Média de tempo de ordenação para 5000 elementos: 7524720 ns
Média de trocas para 5000 elementos: 60173
Média de iterações para 5000 elementos: 4999

Média de tempo de ordenação para 10000 elementos: 55392720 ns
Média de trocas para 10000 elementos: 130762
Média de iterações para 10000 elementos: 9999
```

Figure 4. Resultados da avaliação Shell Sort