

Métodos de Ordenação: Bubble Sort e Quick Sort

Luan Zuffo¹

¹Ciência da Computação – Universidade do Oeste de Santa Catarina (UNOESC)
– Videira – SC – Brasil

luanzuffo@unoesc.edu.br

Abstract. *Sorting methods are ways of ordering vectors in a given order to be able to make searches much faster and more effective. There are several sorting methods, one of the fastest being Quick Sort and one of the slowest, but effective, Bubble Sort.*

Resumo. *Os métodos de ordenação são maneiras de ordenar vetores em uma determinada ordem de modo a poder fazer buscas muito mais rápidas e eficazes. Existem diversos métodos de ordenação, sendo um dos mais rápidos o Quick Sort e um dos mais lentos, porém eficaz, o Bubble Sort.*

1. Bubble Sort

O bubble sort trata-se de um meio de ordenação de elementos no qual é implementado de forma simples, porém, em muitas vezes acaba se tornando lento devido ao número de comparações e trocas de números dentro de um vetor. Portanto, o bubble sort, apesar de ser um método fácil de se implementar, pode ocasionar lentidão dependendo do número de elementos.

O bubble sort funciona da seguinte forma, ele compara o elemento i com o elemento $i+1$, caso o elemento i seja maior que o elemento $i+1$, eles trocam de posição (isso quando for ordenar de forma crescente) utilizando uma variável auxiliar, ele faz isso “ n ” vezes, até o vetor estar completamente ordenado.

Foi realizado alguns testes, com um vetor de 1000 elementos, nos quais continham o nome dos trabalhadores, números de horas trabalhadas e o valor pago por hora. Sendo assim, foi ordenando utilizando o Bubble Sort os das seguintes formas: alfabeticamente, número de horas trabalhadas e os que receberam mais naquele mês. Sendo assim, cada teste deu a seguinte média de tempo (o programa rodou 10 vezes a fim de diminuir o tempo onde os próprios processos do sistema operacional interferirem) ficaram os seguintes:

- **Alfabeticamente:** 0.13631980419158934s;
- **Número de horas trabalhadas:** 0.37462830543518066s;
- **Mais bem pagos:** 0.8699666500091553s;

Sendo assim, pode-se perceber que o Bubble sort teve uma média de 0.4603049198786418s para ordenar esse vetor das 3 formas.

2. Quick Sort

O Quick Sort é uma das formas de ordenação mais rápidas que temos na maioria dos casos, ele provavelmente também é um dos mais utilizados. Dessa forma, ele tem a ideia básica de dividir o problema em dois menores, dessa forma os menores são ordenados de forma independente. Ao final, junta-se os dois vetores e se obtém um vetor ordenado.

O Quick Sort funciona da seguinte forma, tomando como exemplo o nosso teste, vetor de 1000 elementos, é pego um elemento pivô, nesse caso pegamos sempre o último elemento do vetor, porém, poderia ser qualquer outro, é estabelecido um início e um fim do vetor, para assim poder dividi-lo em duas partes. Após isso, é comparado o elemento pivô com o elemento i , caso o elemento i seja maior que o pivô, ele passa para a direita, caso seja menor, fica à esquerda. Porém, nesse exemplo, altera-se apenas a posição do pivô no final da ordenação, para evitar muitas mudanças.

Sendo assim, foram realizados os mesmos testes que o Bubble Sort, ordenando o mesmo vetor no qual contém os nomes dos funcionários, número de horas trabalhadas e valor pago por horas. Portanto, foram realizados os testes de ordenar alfabeticamente, pelo número de horas trabalhadas e o funcionários que mais receberam naquele mês. Os tempos médios (rodado também 10 vezes) de cada ordenação ficou:

- **Alfabeticamente:** 0.002125668525695801s;
- **Número de horas trabalhadas:** 0.005166769027709961s ;
- **Mais bem pagos:** 0.009443330764770507s;

Sendo assim, pode-se perceber que o Quick sort teve uma média de 0.00557858943939209s para ordenar esse vetor das 3 formas.

3. Comparações

Dessa forma, pode-se perceber que o método Quick Sort foi bem mais rápido que o Bubble Sort nesta situação, contudo, ele foi mais difícil de ser aplicado.

A média dos dois foi de 0.4603049198786418s para o método Bubble Sort e de 0.00557858943939209s para o quick sort.

Nestes testes não foram levados em conta o período de leitura e gravação do arquivo CSV no sistema, apenas o tempo de ordenação dos dois métodos.

- [Link repositório](#)
- [Video no youtube](#)