

< VOLTAR



Ordenação de vetores

Apresentar os conceitos de ordenação de vetores e, em especial, o método de ordenação Bubble Sort.

NESTE TÓPICO

- > Introdução
- > Ordenação por Troca (Bolha – BubbleSort)
- > Referências



Introdução

Não se pode considerar uma base de dados apenas como um repositório de dados, pois seria de pouca utilidade armazenar dados que não podem ser facilmente encontrados. Para trazer ordem ao caos foram criados os algoritmos de ordenação e de pesquisa.

Os algoritmos de ordenação trazem a oportunidade de ordenar os dados, para que se tenha resposta às perguntas do tipo: "quais os clientes que mais compraram esse ano?", "quais são os funcionários que tem maior salário?", entre outras.

A ordenação de dados é de extrema importância em um banco de dados. Por isso, tome o seguinte banco de dados apresentado pela Tabela na Figura a seguir.

Cliente	Vendas (R\$)
A	12.856
B	13.423

Cliente	Vendas (R\$)
C	17.366
D	15.352
E	13.432
F	17.382
G	17.432
H	12.423
I	10.324
J	13.312
K	14.353
L	17.435

Todos os dados deste banco são semelhantes e nenhum deles nos salta à vista. De que maneira seria realizada a ordenação destes dados?

Uma alternativa imediata que se pode pensar é percorrer toda a lista e colocar o maior dado no topo. Depois repetir esse mesmo processo, e de novo, e tantas vezes quantas forem o número de registros da lista.

Como será visto a seguir, no entanto, esse método pode ser bastante ineficiente. A eficiência de um método ou algoritmo de ordenação pode ser medida por duas variáveis: o número de comparações e o número de trocas e quanto maior forem esses números, mais ineficiente é o algoritmo.

A lista acima, por exemplo, tem 12 registros. Como se tem que passar 12 vezes pela lista fazendo uma comparação para cada registro chega-se a um total de 144 comparações, e de, no máximo, 144 trocas. Não parece muito, considerando que os computadores hoje podem fazer milhares de operações por segundo, como mostra o exemplo pela Tabela na Figura abaixo.

Registros	Comparações
5	25
25	625
500	250.000
2.000	4.000.000
5.0000	2.500.000.000

Existem três métodos gerais de ordenação: por troca, por seleção e por inserção. Neste tópico, será dada ênfase ao método de ordenação por troca.

Ordenação por Troca (Bolha – BubbleSort)

A ordenação por troca é o método mais básico de ordenação, e consiste em dois laços aninhados. Para facilitar considere os laços como *laço i* e *laço j*. O *laço j* é o interno e ele passa por cada item do vetor (do fim para o início) comparando o elemento da posição (*j*) com o próximo (*j+1*). Caso seja maior que o próximo, estes dois elementos são trocados. Após *j* ter passado por toda o vetor, o valor de *i* é incrementado e essa operação é repetida tantas vezes quantas sejam o número de elementos do vetor.

A seguir é apresentada a função com o método de ordenação por troca.

```
1. void Bubble_Sort (int vet[], int tam){  
2.  
3.     int i, j;  
4.     int aux;  
5.  
6.     for (i=1; i<tam;i++){  
7.         for (j=0;j<tam-i;j++){  
8.             if (vet[j] > vet[j+1]){  
9.                 aux = vet[j];  
10.                vet[j] = vet[j+1];  
11.                vet[j+1] = aux;  
12.            }  
13.        }  
14.    }  
15. }
```

IMPORTANTE

- *i* determina o número de etapas para a ordenação.
- *j* determina o número de comparações em cada etapa e os índices a serem pesquisados para a comparação.

O número máximo de execuções do trecho do algoritmo para que o vetor fique ordenado é $N-1$ vezes, onde N é o número de elementos do vetor.

Abaixo segue o programa completo para o Bubble Sort ordenar um vetor com 5 números inteiros.

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.
4.
5.  void Bubble_Sort (int vet[], int tam){
6.
7.      int i, j;
8.      int aux;
9.
10.     for (i=1; i<tam;i++){
11.         for (j=0;j<tam-i;j++){
12.             if (vet[j] > vet[j+1]){
13.                 aux = vet[j];
14.                 vet[j] = vet[j+1];
15.                 vet[j+1] = aux;
16.             }
17.         }
18.     }
19. }
20.
21. main(){
22.     int vet[5] = {6,8,4,1,7};
23.     int tam = 5;
24.
25.     Bubble_Sort (vet, tam);
26.
27.     for (int i=0;i<tam;i++)
28.         printf("\n vet[%d] = %d",i,vet[i]);
29.
30.     system ("Pause");
31. }
32.
33.
```

Abaixo segue uma animação para exemplificar melhor o método Bubble Sort.



Apresentar a Simulação do método Bubble Sort para ordenar cartas de um baralho

Fonte: <https://www.youtube.com/watch?v=HHVxkJJfwxY>

Quiz

Exercício

Ordenação de vetores

INICIAR ➤

Quiz

Exercício Final

Ordenação de vetores

INICIAR ➤

Referências

MIZRAHI, V. V. *Treinamento em linguagem C*, São Paulo: Pearson, 2008.

SCHILDT, H. C – *Completo e Total*, São Paulo: Pearson, 2006.



Avalie este tópico



ANTERIOR

Passagem de parâmetros



Índice

Biblioteca
(https://www.uninove.br/conheca-
a-
uninove/biblioteca/sobre-
a-
biblioteca/apresentacao/)
Portal Uninove
(http://www.uninove.br)
Mapa do Site

Ajuda?
(https://ava.un
idCurso=)

Pesquisa em vídeo

© Todos os direitos reservados

