

Estrutura de dados

Método Bubble sort

Método de ordenação que compara de forma sucessiva pares de elementos, trocando-os de posição quando se apresentam fora da ordem desejada.

Denomina-se "método da bolha", pois adota a estratégia de "borbulhar" o maior elemento (de maior valor) para o final do arranjo

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[4]	V[3]
1	9	7	1	5
	7	9	1	5

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	9	5

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	9	5

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

O maior elemento, 9, já está na sua posição final.

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

O segundo maior elemento, 7, já está na sua posição final.

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

Iteração	V[0]	V[1]	V[2]	V[3]
3	1	5	7	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

Iteração	V[0]	V[1]	V[2]	V[3]
3	1	5	7	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9

Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

Iteração	V[0]	V[1]	V[2]	V[3]
3	1	5	7	9

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9
Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

Iteração	V[0]	V[1]	V[2]	V[3]
3	1	5	7	9

O terceiro maior elemento, 5, já está na sua posição final.

Estrutura de dados

Método Bubble sort

Iteração	V[0]	V[1]	V[2]	V[3]
1	9	7	1	5
	7	9	1	5
	7	1	5	9
Iteração	V[0]	V[1]	V[2]	V[3]
2	7	1	5	9
	1	7	5	9
	1	5	7	9

Iteração	V[0]	V[1]	V[2]	V[3]
3	1	5	7	9

Consequentemente, o 1 já está na sua posição final.

Estrutura de dados

Método Bubble sort – versão 1

```
...  
int vet[10];  
void bolha (int n){  
    int i, j;  
    for (i=0; i<n; i++){  
        for (j=0; j < n-1; j++)  
            if (vet[j] > vet[j+1])  
                troca(vet, j, j+1);  
    }  
}
```

```
void troca(int *vet, int p1, int p2){  
    int aux;  
    aux= vet[p1];  
    vet[p1]= vet[p2];  
    vet[p2]=aux;  
}
```

Estrutura de dados

Método Bubble sort versão 2 - melhorada

```
void bolha2(int n){
    int i, trocou, aux, fim = n-1;
    do{
        trocou = 0;
        for (i=0; i<fim;i++){
            if(vet[i]>vet[i+1]){
                troca(vet, i, i+1);
                trocou = 1;
            }
        }
        fim--;
    }while(trocou!=0);
}
```

```
void troca(int *vet, int p1, int p2){
    int aux;
    aux= vet[p1];
    vet[p1]= vet[p2];
    vet[p2]=aux;
}
```