

Ordenação e Busca

**Análise e Desenvolvimento de
Sistemas**

**Algoritmos e Estruturas de
Dados I**

19/05/2020

Algoritmo da Bolha ou *Bubble Sort*

- O *Bubble Sort* (Método da Bolha), ou ordenação por flutuação, é um algoritmo de ordenação de vetores que utiliza a mais simples das técnicas;
- A ideia é percorrer o vetor diversas vezes, e a cada passagem fazer *flutuar* para o topo o maior elemento da sequência;
- Essa movimentação lembra a forma como as bolhas em um tanque de água procuram seu próprio nível, daí o nome do algoritmo.



Ordenação por Troca - *Bubble Sort*

- São efetuadas comparações entre os dados armazenados em um vetor de tamanho n ;
- Cada elemento da posição i é comparado com o elemento da posição $i+1$;
- Quando a ordenação desejada (crescente ou decrescente) é encontrada, uma troca de posições entre os elementos é feita;
- Assim, um laço com a quantidade de elementos do vetor é executado e, dentro deste, outro laço que percorre da 1ª à penúltima posição do vetor.

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

x[5], n, i, aux: inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 5$; $i \leftarrow i + 1$)

{ Imprima "Digite o Elemento: ";
leia (x[i]);

para ($n \leftarrow 0$; $n < 5$; $n \leftarrow n + 1$)

{ para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)
{ se ($x[i] > x[i+1]$)
então { aux = x[i];
x[i] = x[i+1];
x[i+1] = aux;

para ($i \leftarrow 0$; $i < 5$; $i \leftarrow i + 1$)

{ imprima "O elemento: ", x[i];

FIM.

Análise da complexidade

- Nesse algoritmo o fator relevante que determina seu tempo de execução é o número de comparações realizadas;
- Considerando que o algoritmo foi implementado para um vetor com 5 posições, verifica-se que o nr de iterações do 1º laço é 5;
- O 2º laço possui 4 iterações, mas como está interno ao 1º, este será executado 20 vezes (5×4);
- Logo, o nr de comparações realizadas é 20.

Gráfico de desempenho – Bubble Sort

Quantidade de trocas

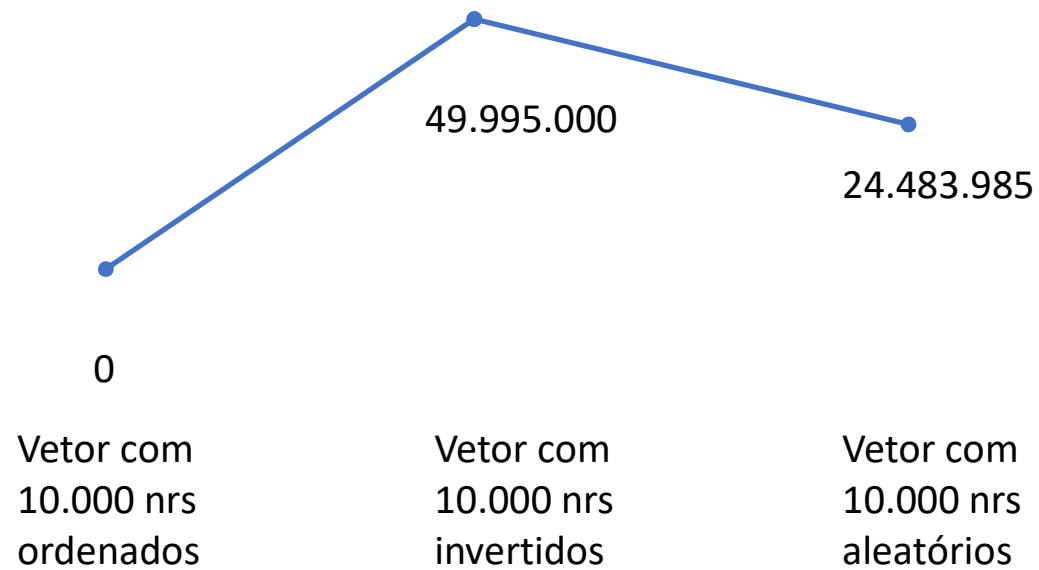


Gráfico de desempenho - Bubble Sort (cont)

Quantidade de Comparações

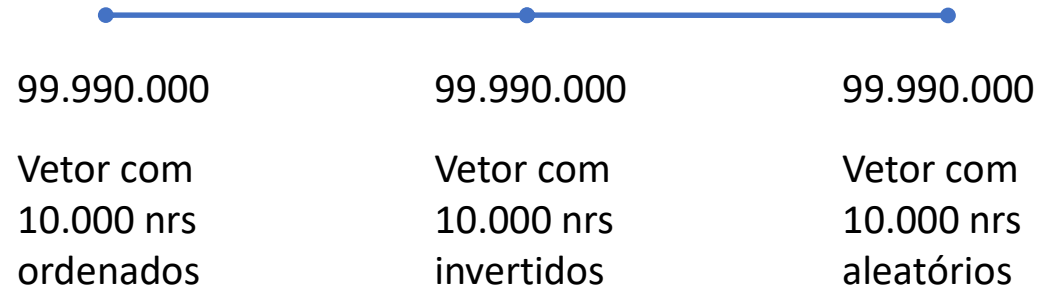
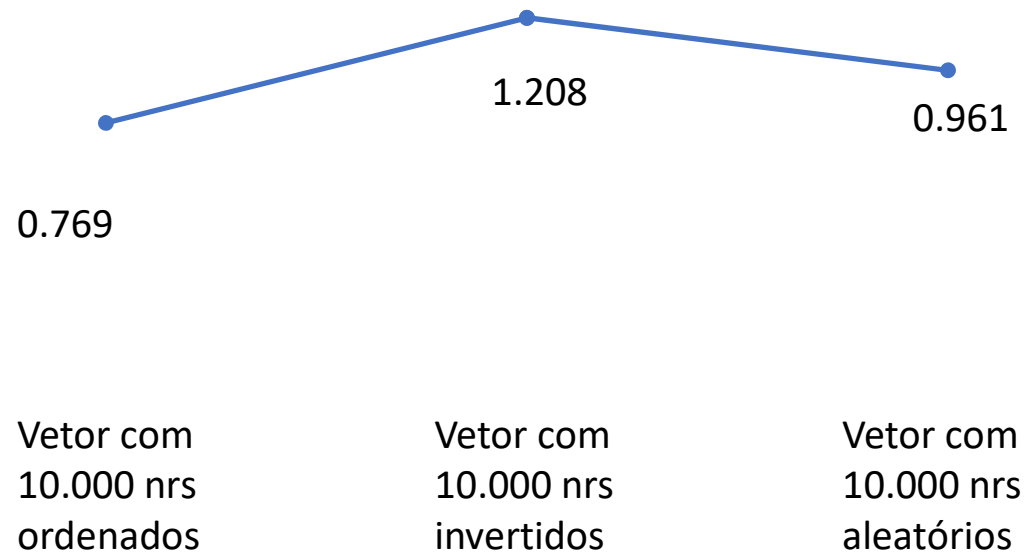


Gráfico de desempenho - Bubble Sort (cont)

Tempo de execução em segundos



Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

```
x[4], n, i, aux: inteiro;
```

INÍCIO

```
para (i ← 0; i < 4; i ← i + 1)
```

```

{ Imprima "Digite o Elemento: ";
  leia (x[i]);
}

```

```
para (n ← 0; n < 4; n ← n + 1)
```

```
para (i ← 0; i < 3; i ← i + 1)
```

```
se (x[i] > x[i+1])
```

então $\left\{ \begin{array}{l} \text{aux} = x[i]; \end{array} \right.$

```
x[i] = x[i+1];
```

```
x[i+1] = aux;
```

4	7	1	5
0	1	2	3



n	aux
0	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

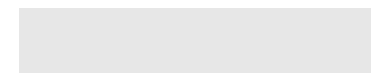
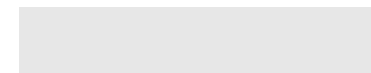
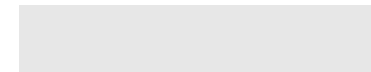
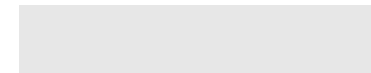
para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	7	1	5
0	1	2	3



n	aux
0	7



Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

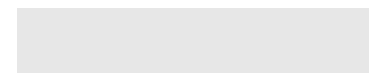
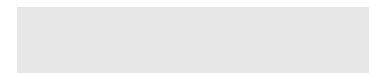
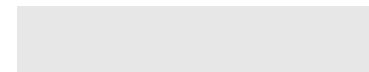
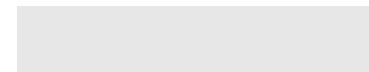
para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	1	7	5
0	1	2	3



n	aux
0	7



Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

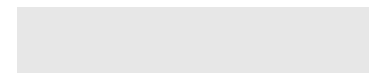
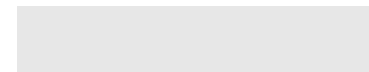
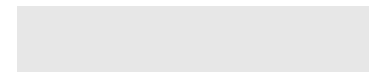
para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	1	7	5
0	1	2	3



n	aux
0	7



Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	1	7	5
0	1	2	3



n	aux
0	7
	7

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	1	5	7
0	1	2	3



n	aux
0	7
	7

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO


para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	1	5	7
0	1	2	3



n	aux
0	7
	7
1	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

4	1	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

Imprima "Digite o Elemento: ";

leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

se ($x[i] > x[i+1]$)

então $aux = x[i]$;

$x[i] = x[i+1]$;

$x[i+1] = aux$;

1	4	5	7
0	1	2	3

↑

n	aux
0	7
	7
1	
	4

Vetor
ordenado!



Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO


para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

 se ($x[i] > x[i+1]$)

 então

$aux = x[i]$;

$x[i] = x[i+1]$;

$x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

 se ($x[i] > x[i+1]$)

 então $aux = x[i]$;

$x[i] = x[i+1]$;

$x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);


para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

 se ($x[i] > x[i+1]$)

 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	
3	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

 se ($x[i] > x[i+1]$)

 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	
3	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

 se ($x[i] > x[i+1]$)

 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	
3	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO

para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)

 se ($x[i] > x[i+1]$)

 então $aux = x[i]$;

$x[i] = x[i+1]$;

$x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	
3	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO


para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	
3	

Algoritmo *Bubble Sort* de ordenação crescente

Algoritmo bolha;

var

$x[4]$, n , i , aux : inteiro;

INÍCIO


para ($i \leftarrow 0$; $i < 4$; $i \leftarrow i + 1$)

{
 Imprima "Digite o Elemento: ";
 leia ($x[i]$);

para ($n \leftarrow 0$; $n < 4$; $n \leftarrow n + 1$)

{
 para ($i \leftarrow 0$; $i < 3$; $i \leftarrow i + 1$)
 {
 se ($x[i] > x[i+1]$)
 então {
 $aux = x[i]$;
 $x[i] = x[i+1]$;
 $x[i+1] = aux$;

1	4	5	7
0	1	2	3



n	aux
0	7
	7
1	
	4
2	
3	
4	

- Implementar em C++ o algoritmo *Bubble Sort*:
 - Crescente e
 - Decrescente.

Bubble sort

Array

6	3	0	5	1
---	---	---	---	---

FIM