

# Bubble sort

O ***bubble sort***, ou ordenação por flutuação (literalmente "por bolha"), é um algoritmo de ordenação dos mais simples. A ideia é percorrer o vector diversas vezes, a cada passagem fazendo 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, e disso vem o nome do algoritmo.

No melhor caso, o algoritmo executa  $n^2/2$  operações relevantes, onde  $n$  representa o número de elementos do vector. No pior caso, são feitas  $2n^2$  operações. No caso médio, são feitas  $5n^2/2$  operações. A complexidade desse algoritmo é de Ordem quadrática. Por isso, ele não é recomendado para programas que precisem de velocidade e operem com quantidade elevada de dados.

O algoritmo pode ser descrito em pseudo-código como segue abaixo. **V** é um VECTOR de elementos que podem ser comparados e **n** é o tamanho desse vector.

```
BUBBLESORT (V[], n)
1   houveTroca <- verdade           # uma variável de controle
2   enquanto houveTroca for verdade faça:
3       houveTroca <- falso
4       para i de 1 até n-1 faça:
5           se V[i] vem depois de V[i + 1]
6               então troque V[i] e V[i + 1] de lugar e
7               houveTroca <- verdade
```

## Assembly

```
/*void bubble_as2 (int *x, int n);*/
.globl bubble_as2
/* assembler utilizado gas (x86 - Linux) */
bubble_as2:
    pushl %ebp
    movl %esp, %ebp
    movl 12(%ebp), %eax /* tamanho -> %eax */
    movl 8(%ebp), %ecx /* início do vetor -> %ecx */
    movl $4, %ebx
    dec %eax
    mul %ebx
    addl %eax, %ecx /* %ecx aponta p/ o último do elemento do vetor */
    pushl %ecx
_bubble_as2_l1:
    movl $0, %edx
    movl 8(%ebp), %ebx
    movl %ebx, %eax
    addl $4, %eax
_bubble_as2_l2:
    cmp %eax, %ecx
    jl _bubble_as2_l1_end
    movl (%ebx), %ecx
    cmp (%eax), %ecx
    jl _bubble_as2_l2_end
```

```
    /* troca */
    movl (%eax), %edx
    movl %edx, (%ebx)
    movl %ecx, (%eax)
    movl $1, %edx
_bubble_as2_l2_end:
    movl %eax, %ebx
    addl $4, %eax
    movl (%esp), %ecx
    jmp _bubble_as2_l2
_bubble_as2_l1_end:
    cmp $0, %edx
    je _bubble_as2_fim
    popl %ecx
    subl $4, %ecx
    pushl %ecx
    jmp _bubble_as2_l1
_bubble_as2_fim:
    leave
    retts
```

## C

```
#include <stdbool.h>

inline void troca(int* a, int* b)
{
    int aux = *a;
    *a = *b;
    *b = aux;
}

void bubbleSort (int *primeiro, int *ultimo)
{
    bool naoTrocou;
    int *posAtual;
    for (; ultimo > primeiro; --ultimo)
    {
        naoTrocou = true;
        for (posAtual = primeiro; posAtual < ultimo; ++posAtual)
        {
            if (*posAtual > *(posAtual+1))
            {
                troca (posAtual, posAtual+1);
                naoTrocou = false;
            }
        }
    }
    if (naoTrocou) return;
```

```
    }  
}  
  
public class Bolha {  
  
    public void bubbleSort(int v[]) {  
  
        for (int i = v.length - 1; i >= 1; i--)  
        {  
            for (int j = 1; j <= i; j++) {  
                if (v[j - 1] > v[j]) {  
  
                    int aux = v[j];  
                    v[j] = v[j - 1];  
                    v[j - 1] = aux;  
  
                }  
            }  
        }  
    }  
}
```

## C

Usando "do while".

```
void swapbubble( int v[], int i)  
{  
  
    int aux=0;  
  
    aux=v[i];  
    v[i] = v[i+1];  
    v[i+1] = aux;  
  
}  
  
void bubble(int v[], int qtd)  
{  
    int i;  
    int trocou;  
  
    do  
    {  
        qtd--;  
        trocou = 0;  
  
        for(i = 0; i < qtd; i++)  
            if(v[i] > v[i + 1])  
            {  
                swapbubble(v, i);  
                trocou = 1;  
            }  
    }  
    while(trocou == 1);  
}
```

```
        trocou = 1;

    }

    }while(trocou);
}
```

### Método de ordenação Bolha com ordenação de strings.

```
void bubble(int v[], int qtd)
//UTILIZA BIBLIOTECA string.h
{
    int i;
    int trocou;
    char aux;

    do
    {
        qtd--;
        trocou = 0;

        for(i = 0; i < qtd; i++)
            if(strcasecmp(v[i],v[i + 1])>0)
            {
                /*o ideal seria fazer uma função troca aqui*/
                strcpy(aux, v[i]);
                strcpy(v[i], v[i + 1]);
                strcpy(v[i + 1], aux);
                trocou = 1;
            }
    }while(trocou==1);
}
```

### C++

```
#include "Bubblesort.h"
void bubblesort(vetor &a, tamanho n)
{
    for(indice j= n-1; j>0; j--){
        for(i=0;i<j;i++){
            if(a[i+1] < a[i]){
                swap(a[i+1], a[i]);
            }
        }
    }
}
```

```
for (int i = vetor.length - 1; i > 0; i--)
{
    for (int j = 0; j < i; j++)
```

```

        {
            if (vetor[i] < vetor[j])
            {
                int swap  = vetor[i];
                vetor[i]   = vetor[j];
                vetor[j] = swap;
            }
        }
    }
}

```

## ML

[http://pt.wikipedia.org/skins-1.5/common/images/button\\_hr.png](http://pt.wikipedia.org/skins-1.5/common/images/button_hr.png)

```

fun fix ( f,x) =
    let val fx = f(x)
    in
        if x = fx then x
        else fix(f,fx)
    end;

fun bubble ([]) = []
  | bubble([a]) = [a]
  | bubble(a::b::x) =
      if a <= b then a::bubble(b::x)
      else b::bubble(a::x);

fun bubblesort( lista ) = fix (bubble,lista);

```

## Pascal

O código a seguir ilustra o algoritmo, para ordenar n números inteiros:

```

program ordenacao;
uses crt;
const
    n = 20;
var
    vet:array[1..n]of integer;
    i,j,aux: integer;
begin
    randomize;
    {Preenche o array com valores aleatórios}
    for i := 1 to n do
        vet[i] := random(100);
    {Ordena o array}
    for i := n downto 2 do
        for j := 1 to i-1 do
            if vet[j] > vet[j+1] then

```

```
begin
    aux := vet[j];
    vet[j] := vet[j+1];
    vet[j+1] := aux;
end;
end.
```

### Implementação do Algoritmo Bolha com FLaG

```
program BolhacomFLag;
uses wincrt;
const
n=10;
var
vet:array[1..n] of integer;
i,aux:integer;
houveTroca:boolean;
begin
i:=0;
houveTroca:=true;
randomize;
for i:=1 to n do
    vet[i]:=random(100);
    for i:=1 to n do
        writeln(vet[i]);
    repeat
        houveTroca:=false;
        for i:=1 to n-1 do
            if ( vet[i] > vet[i+1]) then
                begin
                    aux := vet[i];
                    vet[i]:= vet[i+1];
                    vet[i+1]:= aux;
                    houveTroca:=true;
                end;
            until (houveTroca = false);
        writeln('Escrevendo Vetor Ordenado');
        for i:=1 to n do
            writeln(vet[i]);
        {by X}
    end.
```

### Python

```
def bubblesort(l):
    for passesLeft in range(len(l)-1, 0, -1):
        for index in range(passesLeft):
            if l[index] < l[index + 1]:
                l[index], l[index + 1] = l[index + 1], l[index]
```

```
        return l

def bubbleSort(L,n):
    flag = True
    while flag:
        flag = False
        for i in range(n-1):
            if L[i] > L[i+1]:
                L[i],L[i+1] = L[i+1],L[i]
                flag = True
```

## Perl

```
sub swap {
    @_ [0, 1] = @_ [1, 0];
}

sub bubble_sort {
    for ($i=$|; $i < $#_; ++$i) {
        for ($j=$|; $j < $#_; ++$j) {
            ($_[ $j] > $ _ [ $j+1]) and swap($_ [ $j], $_ [ $j+1]);
        }
    }
}
```

## C#

```
private void BubbleSort(int[] vetor)
{

    //Ordem Decrescente

    for (int i = vetor.Length - 1; i > 0; i--)
    {
        for (int j = 0; j < i; j++)
        {
            if (vetor[i] > vetor[j])
            {
                int swap    = vetor[i];
                vetor[i]    = vetor[j];
                vetor[j] = swap;
            }
        }
    }
}
```

## PHP

```
<?php

    /*
        Esta função troca o valor de duas variáveis entre si.
        Opcional. Pode ser embutido na bolha
    */
    function swap(&$valor_1, &$valor_2) {
        list($valor_1, $valor_2) = array($valor_2, $valor_1);
    }

    /* Array de teste */
    $arrToSort = array(1, 4, 7, 3, 8, 9, 10);

    /* a BOLHA! ;- ) */
    for ($i = 0; $i < count($arrToSort); $i++) {
        for ($j = $i; $j < count($arrToSort); $j++) {
            if ($arrToSort[$i] > $arrToSort[$j]) {
                swap($arrToSort[$i], $arrToSort[$j]);
            }
        }
    }

    /* Opcional. Exibe o array de um jeito que nós podemos entender! =D */
    print_r($arrToSort);
?>
```

```
<?php
function BubbleSort( $items ) {
    $temp = "";
    $size = count( $items );
    for( $i = 1; $i < $size; $i++ ) {
        for( $j = 1; $j < $size - $i; $j++ ) {
            if( $items[$j+1] < $items[$j] ) {
                $temp = $items[$j];
                $items[$j] = $items[$j+1];
                $items[$j+1] = $temp;
            }
        }
    }
}

$items = array(31, 41, 59, 26, 41, 58);
BubbleSort( $items );
?>
```



---

## Shell script

```
#!/bin/bash
# Criado em: Qua 12/Jul/2006 hs 12:34
# Last Change: Qua 12 Jul 2006 12:57:59 BRT
# Instituicao:
# Proposito do script: algoritmo de ordenação

echo " Digite cinco numeros!"

for ((i=0; i<=4; i++)); do
    read n[i]
done

    for ((i=0; i<=3; i++)); do
        for ((j=i+1; j<=4; j++)) do

            if [ ${n[i]} -gt ${n[j]} ]; then

                x=${n[i]}
                n[i]=${n[j]}
                n[j]=$x
            fi

        done
    done

echo "Lista ordenada!"
for ((i=0; i<=4; i++)); do
    echo digitado ${n[i]}
done
```

## Lua

```
function bubbleSort(v)
    for i=#v, 1, -1 --Lembrando que vetores no lua começam no "1" e não no "0"
    do
        for j=1, i-1, 1
        do
            if(v[j]>v[j+1])
            then
                v[j],v[j+1] = v[j+1],v[j]
            end
        end
    end
end
```

---

## Matlab

```
for(i = 1:n)
    for(j = 1:n-1)
        if(x(j) > x(j + 1))
            aux = x(j);
            x(j) = x(j + 1);
            x(j + 1) = aux;
        end
    end
end
```

## ASP

```
<%
Dim links(5)
links(0) = 1
links(1) = 3
links(2) = 5
links(3) = 2
links(4) = 4

'Esta função troca o valor de duas variáveis entre si.
'Opcional. Pode ser embutido na bolha
function swap(i, j)
    valor_1_antigo = links(i)
    valor_2_antigo = links(j)

    links(i) = valor_2_antigo
    links(j) = valor_1_antigo
end Function

'A BOLHA
for i = 0 to UBound(links)
    for j = i+1 to UBound(links)
        if (links(i) > links(j)) then
            call swap(i, j) 'Passa o ponteiro para a funcao swap
        end if
    next
next

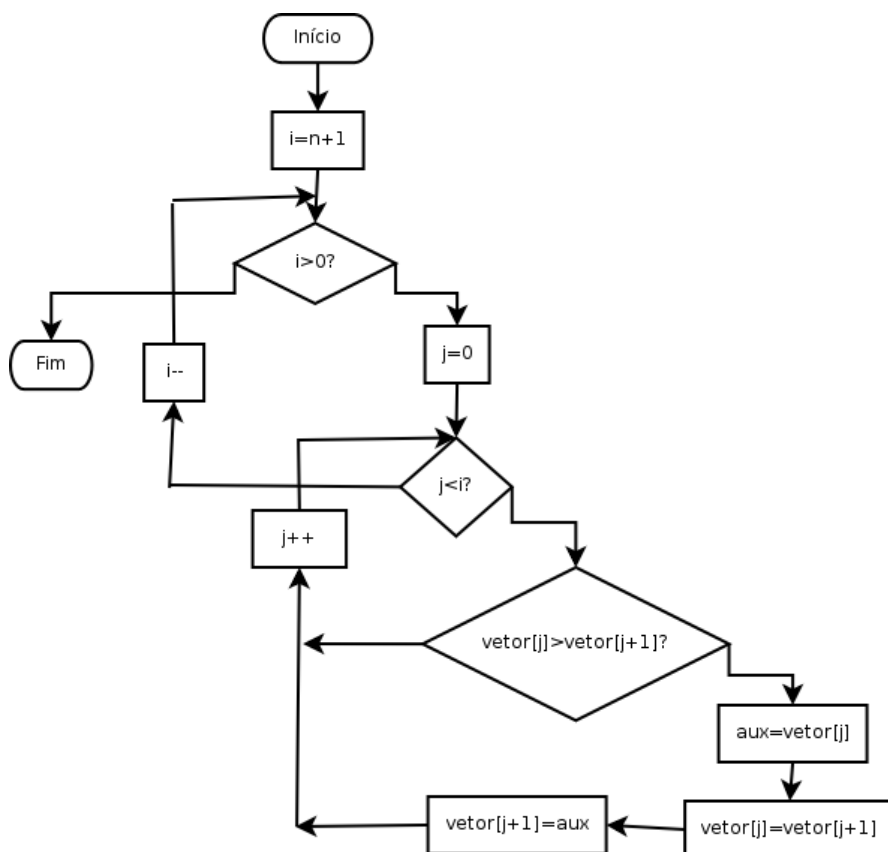
'Resultado
for i = 0 to UBound(links)
    Response.write links(i) & " "
next
%>
```

## Visual Basic

```
Private Sub sbOrdena(aNumbers() As Integer)
    Dim iNumber1 As Integer
    Dim iNumber2 As Integer
    Dim iNumberAux As Integer

    For iNumber1 = 0 To UBound(aNumbers)
        For iNumber2 = 0 To UBound(aNumbers) - 1
            If aNumbers(iNumber2) > aNumbers(iNumber2 + 1) Then
                iNumberAux = aNumbers(iNumber2)
                aNumbers(iNumber2) = aNumbers(iNumber2 + 1)
                aNumbers(iNumber2 + 1) = iNumberAux
            End If
        Next iNumber2
    Next iNumber1
End Sub
```

## Fluxograma



## Simulador Online

- Para um entendimento mais detalhado, utilize o simulador online <sup>[1]</sup> para visualizar o método Bubble Sort ordenando uma sequência de números aleatórios.

## Ver também

- Ordenação de vector
- Quick sort
- Heapsort
- Merge sort
- Selection sort
- Pesquisa binária

## Ligações externas

- C2.com <sup>[2]</sup>
- CS <sup>[3]</sup>
- Bubble Código de classificação <sup>[4]</sup>
- [5]

## Referências

- [1] <http://wikipedia.artudi.org/Bubble%20Sort.php>  
[2] <http://c2.com/cgi/wiki?BubbleSort>  
[3] <http://www.cs.ubc.ca/spider/harrison/Java/sorting-demo.html>  
[4] [http://www.algorithm-code.com/wiki/Bubble\\_sort](http://www.algorithm-code.com/wiki/Bubble_sort)  
[5] [http://en.wikipedia.org/wiki/Sorting\\_algorithm#Insertion\\_sort](http://en.wikipedia.org/wiki/Sorting_algorithm#Insertion_sort)

# Fontes e Editores da Página

**Bubble sort** *Fonte:* <http://pt.wikipedia.org/w/index.php?oldid=20289088> *Contribuidores:* Adailton, Andrevruas, Aprendiz de feiticeiro, Carnevalli, Ccuembej, EduM, Eduardocolabardini, Educobuci, Felipo bacani, Gdst, Hgferman, João Carvalho, Kaze Senshi, Leonardo.stabile, LeonardoG, Luís Felipe Braga, Marivb, Mestre5, Nuno Tavares, Níssius Guilet Ribas, OffsBlink, Rafael.afonso, Rafaelcosta1984, Reynaldo, Santana-freitas, Steinh, Teles, Thiagoharry, Tyran, Xandi, 140 edições anónimas

# Fontes, licenças e editores da imagem

**Ficheiro:Bubblesort.png** *Fonte:* <http://pt.wikipedia.org/w/index.php?title=Ficheiro:Bubblesort.png> *Licença:* Public Domain *Contribuidores:* Original uploader was Thiagoharry at pt.wikipedia

# Licença

---

Creative Commons Attribution-Share Alike 3.0 Unported  
<http://creativecommons.org/licenses/by-sa/3.0/>

---