



## Ordenação de Vetores - Programação II

**Autor** Prof.: Delcino Picinin Júnior

### Bubble Sort

Este algoritmo é um dos mais simples levando em conta a implementação, apesar de não ter um bom desempenho se comparado a outros tipos de algoritmos de ordenação

A ideia é sempre iterar toda a lista de itens quantas vezes forem necessário até que os itens estejam na ordem correta. Durante a iteração, o algoritmo deve comparar e ordenar pares de valores: O item que se está iterando e o seu vizinho à direita.

Processo 1: [4, 5, 3, 2, 1]

Processo 2: [4, 3, 5, 2, 1]

Processo 3: [4, 3, 2, 5, 1]

Processo 4: [4, 3, 2, 1, 5]

Processo 5: [3, 4, 2, 1, 5]

Processo 6: [3, 2, 4, 1, 5]

Processo 7: [3, 2, 1, 4, 5]

Processo 8: [3, 2, 1, 4, 5]

Processo 9: [2, 3, 1, 4, 5]

Processo 10: [2, 1, 3, 4, 5]

Processo 11: [2, 1, 3, 4, 5]

Processo 12: [1, 2, 3, 4, 5]

## Código 1: Exemplo do Bubble Sort

---

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void bolha (int *vetor, int n) {
5     int k, j, aux;
6
7     for (k = 1; k < n; k++) {
8         for (j = 0; j < n - k; j++) {
9             if (vetor[j] > vetor[j + 1]) {
10                 aux = vetor[j];
11                 vetor[j] = vetor[j + 1];
12                 vetor[j + 1] = aux;
13             }
14         }
15     }
16 }
17
18 int main () {
19     int i, val[] = { 15, 30, 10, 20, 25 };
20
21     //ordena o array
22     bolha(val,5);
23
24     //mostra os valores do array
25     for( i = 0 ; i < 5; i++ ) {
26         printf ("%i ", val[i]);
27     }
28
29     return (0);
30 }
```

---

## Quicksort

A função `qsort()` é uma função em C utilizada para ordenação de vetores.

A função `comparador()` recebe dois números do vetor e os compara. Veja que a função recebe dois ponteiros `void` e por isso é preciso fazer o cast para o tipo de dado do vetor, no exemplo abaixo, valores inteiros.

Código 2: Exemplo de `scanf`

---

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int comparador(const void *a, const void *b) {
5     return ( *(int*)a - *(int*)b );
6 }
7
8 int comparador2(const void *a, const void *b) {
9     if ( *(int*)a > *(int*)b ) {
10         return 1;
11     } else if ( *(int*)a < *(int*)b ) {
12         return -1;
13     } else {
14         return 0;
15     }
16 }
17
18 int main () {
19     int i, val[] = { 15, 30, 10, 20, 25 };
20
21     //ordena o vetor
22     qsort(val, 5, sizeof(int), comparador);
23
24     //mostra os valores do vetor
25     for( i = 0 ; i < 5; i++ ) {
26         printf ("%i ", val[i]);
27     }
28
29     return (0);
30 }
```

---