

Bubble Sort (Bubbles)

→ Ordenação por Bolha

- compara pares de elementos adjacentes e os troca de lugar se estiverem na ordem errada
- processo se repete até que mais nenhuma troca seja necessária (elementos já ordenados)
- Performance
 - melhor caso : $O(N)$ // dados ordenados
 - pior caso : $O(N^2)$ //
 - Não recomendado para grandes conjuntos de dados

• Algoritmo (otimizado)

```
void bubbleSort (int *v, int N) {
```

```
    int i, continua, aux, fim = N;
```

```
    do {
```

```
        continua = 0;
```

```
        for (i = 0; i < fim - 1; i++) {
```

```
            if (v[i] > v[i+1]) { // compara adjacentes
```

```
                swap(v[i], v[i+1]); // troca valores
```

```
                continua = i;
```

```
            } // if
```

```
        } // for
```

```
        fim--; // último elemento já posicionado corretamente
```

```
    } while (continua != 0);
```

```
}
```

// Carregar o maior elemento até o final do vetor

// diminuir o vetor em 1

// repete o processo até todos estarem em seus lugares.

Exemplo:

Sem Ordenar

23 4 67 -8 90 54 21

• iteração 01:

i = 0	23	4	67	-8	90	54	21	Troca
i = 1	4	23	67	-8	90	54	21	OK
i = 2	4	23	67	-8	90	54	21	Troca
i = 3	4	23	-8	67	90	54	21	OK
i = 4	4	23	-8	67	90	54	21	Troca
i = 5	4	23	-8	67	54	90	21	Troca
Final	4	23	-8	67	54	21	90	

• iteração 02:

i = 0	4	23	-8	67	54	21	90	OK
i = 1	4	23	-8	67	54	21	90	Troca
i = 2	4	-8	23	67	54	21	90	OK
i = 3	4	-8	23	67	54	21	90	Troca
i = 4	4	-8	23	54	67	21	90	Troca
Final	4	-8	23	54	21	67	90	

• iteração 03:

i = 0	4	-8	23	54	21	67	90	troca
i = 1	-8	4	23	54	21	67	90	ok
i = 2	-8	4	23	54	21	67	90	ok
i = 3	-8	4	23	54	21	67	90	troca
Final	-8	4	23	21	54	67	90	

• iteração 04:

i = 0	-8	4	23	21	54	67	90	ok
i = 1	-8	4	23	21	54	67	90	ok
i = 2	-8	4	23	21	54	67	90	troca
Final	-8	4	21	23	54	67	90	

• iteração 05:

i = 0	-8	4	21	23	54	67	90	ok
i = 1	-8	4	21	23	54	67	90	ok
Final	-8	4	21	23	54	67	90	

obs: Não houve mudanças \Rightarrow Ordenação Concluída!

- pseudo código

bubbleSort (vetor[, N])

For $i = 0$ até $i \leq N-1$ $[0 \dots N-2]$

 For $j = N-1$; até $j > i$, $j--$
 • se $data[j] < data[j-1]$:
 | swap($data[j]$, $data[j-1]$)
 // se
 // for
 // for

- Desvantagens:

- borbulha os itens iterativamente
- pior caso $O(n^2)$