



# Estrutura de Dados – I

## Ordenação

Prof. MSc. Rafael Staiger Bressan  
[rafael.bressan@unicesumar.edu.br](mailto:rafael.bressan@unicesumar.edu.br)



# Introdução

- Ordenar
  - Processo de reorganizar um conjunto de objetos em uma ordem ascendente ou descendente.
- A ordenação visa facilitar a recuperação posterior de itens do conjunto ordenado.
  - Imagine um lista de telefone desordenada...



# Ordenação

## Métodos Simples

- Método de Seleção (Selection Sort)
- Método de Inserção (Insertion Sort)
- **Método de Troca (Bubble Sort, ou Método de Flutuação / Bolha)**
- ...



# *Bubble Sort*

- Algoritmo de ordenação simples.
  - a ideia é percorrer o vetor várias vezes (geralmente com o número de elementos), e a cada vez, 'flutuar' o maior elemento da sequência, ou seja, essa movimentação lembra a forma de como as bolhas em um reservatório de água, procuram seu próprio nível.

# *Bubble Sort*

5

4

8

2

1

7

# *Bubble Sort*



**5 > 4 ?**

# *Bubble Sort*



**5 > 4 ?**  
**SIM**

# *Bubble Sort*



**Troca**



# *Bubble Sort*



**5 > 8 ?**

# *Bubble Sort*



**$5 > 8 ?$**

**Não**

# *Bubble Sort*



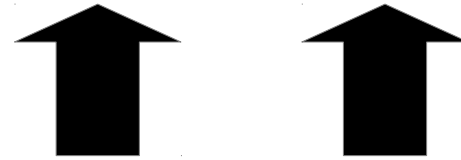
**$8 > 2 ?$**

# *Bubble Sort*



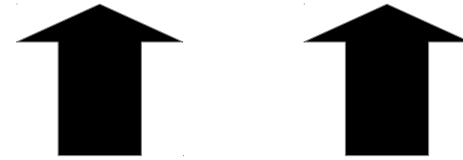
**Troca**

# *Bubble Sort*



**8 > 1 ?**

# *Bubble Sort*



**Troca**

# *Bubble Sort*



**8 > 7 ?**

# *Bubble Sort*



**Troca**



# *Bubble Sort*



**4 > 5 ?**

# *Bubble Sort*



**4 > 5 ?**  
**Não**

# *Bubble Sort*



**$5 > 2 ?$**

# *Bubble Sort*



**Troca**

# *Bubble Sort*



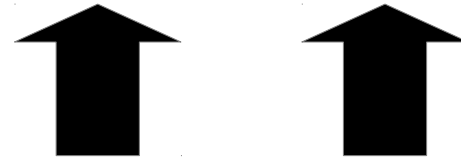
**$5 > 1 ?$**

# *Bubble Sort*



**Troca**

# *Bubble Sort*



**5 > 7 ?**

# *Bubble Sort*

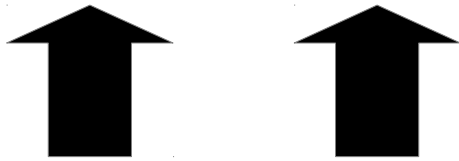


**1 > 7 ?**

**Não**

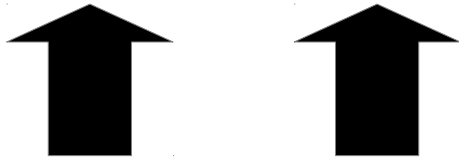


# *Bubble Sort*



**4 > 2 ?**

# *Bubble Sort*



**Troca**

# *Bubble Sort*



**4 > 1 ?**

# *Bubble Sort*



**$4 > 1 ?$**

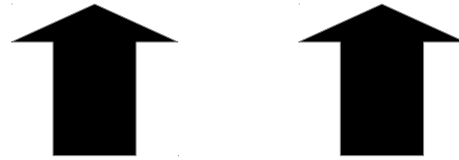
**Sim**

# *Bubble Sort*



**4 > 5 ?**  
**Troca**

# *Bubble Sort*



**4 > 5 ?**

# *Bubble Sort*



**$4 > 5 ?$**

**Não**

# *Bubble Sort*



**2 > 1 ?**



# *Bubble Sort*



**2 > 1 ?**

**Sim**

# *Bubble Sort*



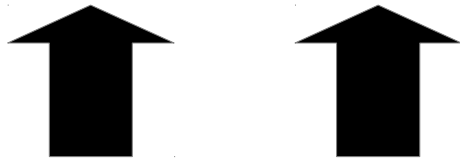
**$2 > 1 ?$**   
**Troca**

# *Bubble Sort*



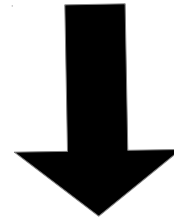
**$2 > 4 ?$**

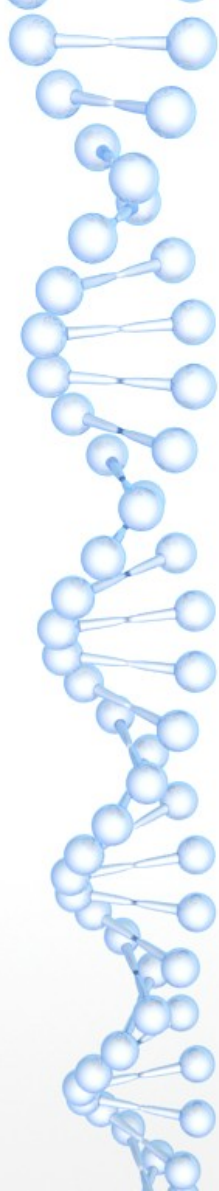
# *Bubble Sort*



**1 > 2 ?**

# *Bubble Sort*





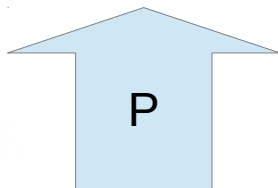
2020

```
1 | #include <stdio.h>
2 | main(){
3 |     int tamanhoVetor, i, vetor[10], controle, aux;
4 |     for (i=0; i<10;i++){
5 |         printf("Vetor[%d] : ", i+1);
6 |         scanf("%d", &vetor[i]);
7 |     }
8 |     tamanhoVetor = 10;
9 |     do{
10 |         controle = 0;
11 |         for (i=0; i<tamanhoVetor-1; i++){
12 |             if (vetor[i] > vetor[i+1]){
13 |                 aux = vetor[i+1];
14 |                 vetor[i+1] = vetor[i];
15 |                 vetor[i] = aux;
16 |                 controle = 1;
17 |             }
18 |         }
19 |         tamanhoVetor--;
20 |     }while(controle != 0);
21 |     for (i=0; i<10;i++){
22 |         printf(" %d ", vetor[i]);
23 |     }
24 | }
```

# Atividade

- Considere a seguinte sequencia dos dados em uma lista simplesmente encadeada.

20	8	1	30	10	6
----	---	---	----	----	---



- Desenvolva uma função em C *“bubbleSort”* para ordenação dos dados de uma lista simplesmente encadeada.