



Universidade Federal Rural de Pernambuco
(UFRPE)
- Unidade Acadêmica de Garanhuns-

Algoritmos de Ordenação

Conceitos Básicos – Bubble Sort

Prof. Priscilla Kelly Machado Vieira

Apresentação do Capítulo

2

- Introdução
- Complexidade
- Algoritmo *in-place*
- Algoritmo Estável
- Bubble Sort
 - Características
 - Análise
- Exercícios

Introdução

3

- Maggie Sort:

<http://www.youtube.com/watch?v=Zybl598sK24>

Ordenação: Relevancia

Resultado: 20

Visualização: Lista Tabela

Páginas: 1 2 3 4 5 6 7 8 ...

de: R\$ 1.699,00
por: R\$ 1.399,00
ou 12x de R\$ 116,58
sem juros no cartão

Comprar

ou

Comprar com 1-Click

[Clique aqui para ativar](#)

Adicionar à lista

Veja a regra

+ Ver similares

Complexidade

4

- Nem todos os algoritmos funcionam bem para um número grande de dados
- Número de passos para executar um algoritmo
 - Máximo
 - Médio
 - Mínimo

Em relação ao tamanho
da entrada (n)

Algoritmo *In place*

5

- A ordenação é realizada no mesmo local onde os dados são armazenados. No máximo precisa de uma memória extra (no máximo é constante)

Algoritmo Estável

6

- Dado dois elementos R e S com a mesma chave, se R aparece antes de S na lista original, R aparecerá antes de S na lista ordenada final
- Importante quando se tem dados satélites

10 (Carlos)	2 (Maria)	7 (Pedro)	7 (Carla)	1 (João)
-------------	-----------	-----------	-----------	----------

Ordenada

1 (João)	2 (Maria)	7 (Pedro)	7 (Carla)	10 (Carlos)
----------	-----------	-----------	-----------	-------------

Algoritmos de Ordenação

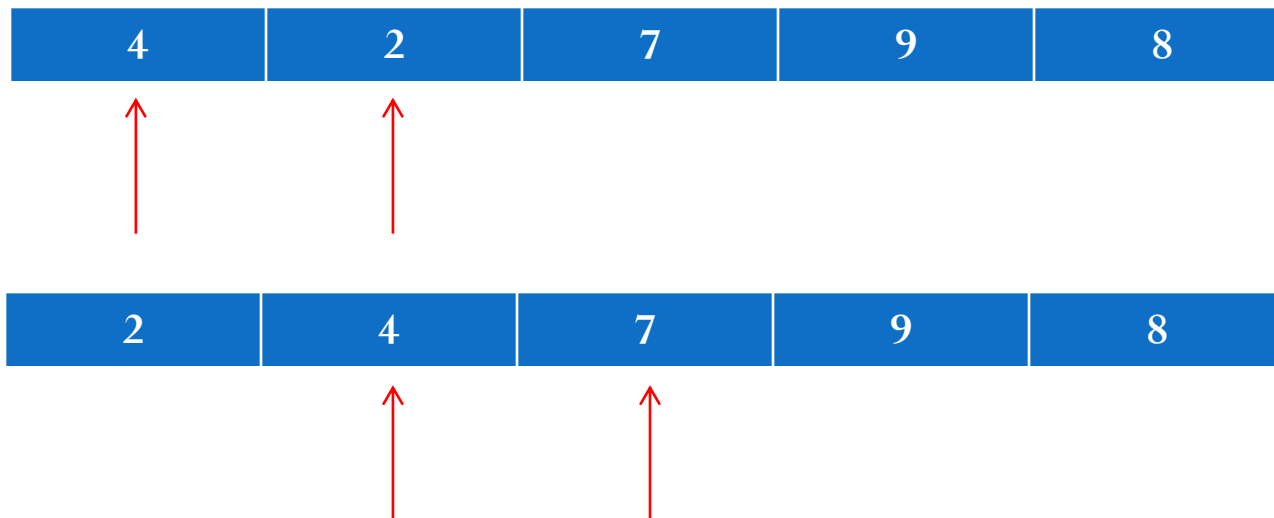
7

- Bubble Sort
- Selection Sort
- Insertion Sort
- Quick Sort
- Merge Sort

Bubble Sort

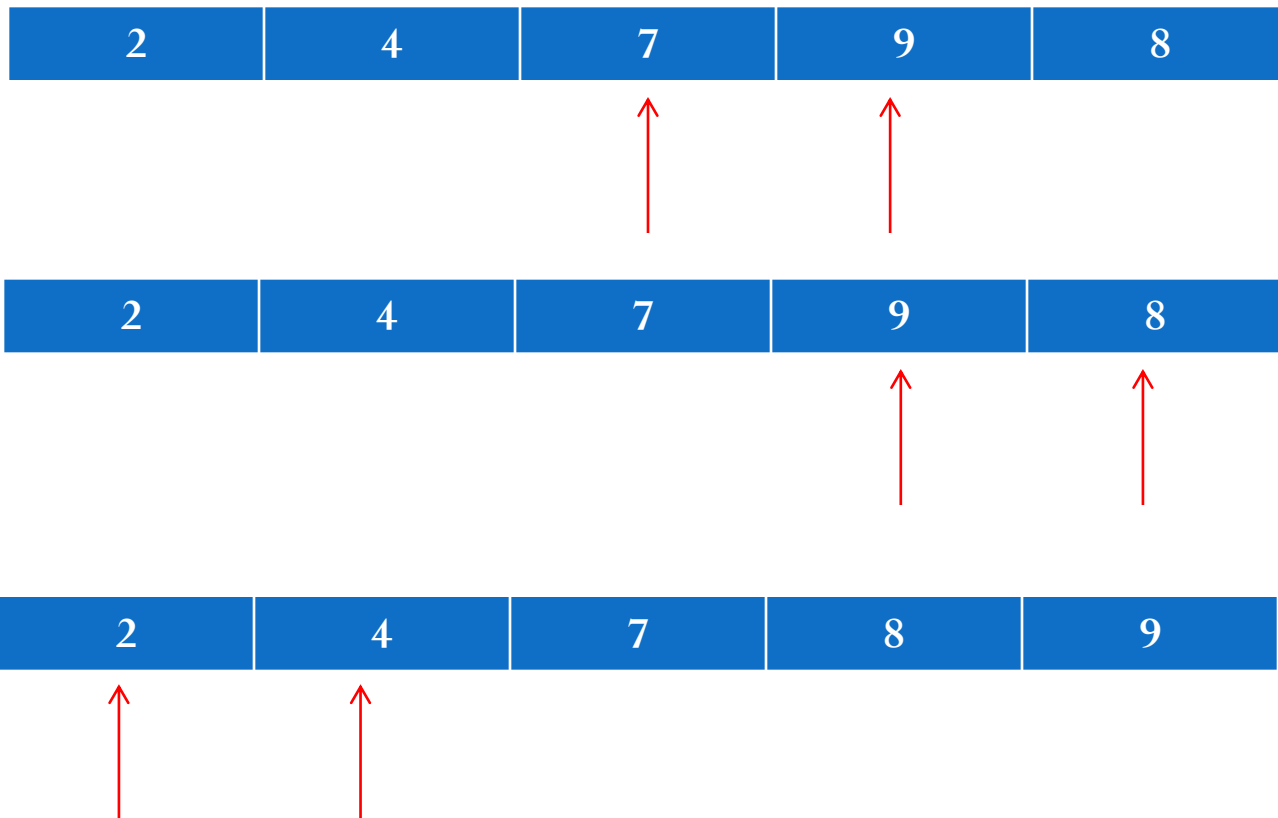
8

- Intuição
 - Percorrer o vetor diversas vezes, a cada passagem “Borbulhando” valores mais elevados (ou mais baixos) para o final (ou para o início) do arranjo



Bubble Sort

9



Bubble Sort

10



...

Bubble Sort

11

- <http://www.site.uottawa.ca/~stan/csi2514/applets/sort/sort.html>
- Algoritmo

```
public static void bubblesort(int[] a){
    for(int i=0; i<a.length-1; i++)
        for(int j=0; j<a.length-i-1; j++)
            if(a[j] > a[j+1])
                a = swap(a, j, j+1);
}

private static int[] swap(int[] a, int i, int j) {
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
    return a;
}
```

Exercício

12

- Ordene as listas utilizando *bubble sort*
 - 9, 12, 2, 3, 8, 0, 2
 - 21, 23, 2, 34, 245, 33, 62

Bubble Sort

13

- Características
 - Implementação simples
 - Fácil entendimento
 - In place
 - Estável
 - Complexidade
 - Bom para entradas pequenas
 - Comparações adjacentes

Bubble Sort

14

- Qual o problema que podemos perceber no algoritmo anterior?

Comparações são realizadas mesmo se a lista estiver ordenada!



Bubble Sort

15

- Algoritmo

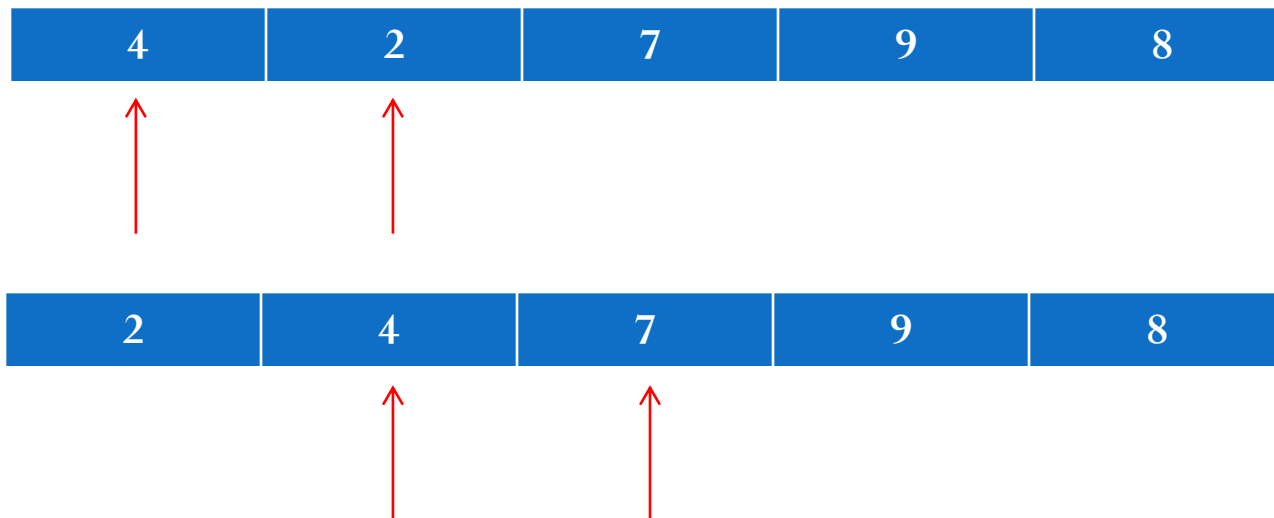
```
public static void b(int[] v){  
    boolean trocou = true;  
    while (trocou) {  
        trocou = false;  
        for (int j = 0; j < v.length-1; j++)  
            if (v[j] > v[j+1]){  
                int aux = v[j];  
                v[j] = v[j+1];  
                v[j+1]=aux;  
                trocou = true;  
            }  
    }  
}
```

O que ele faz?

Bubble Sort

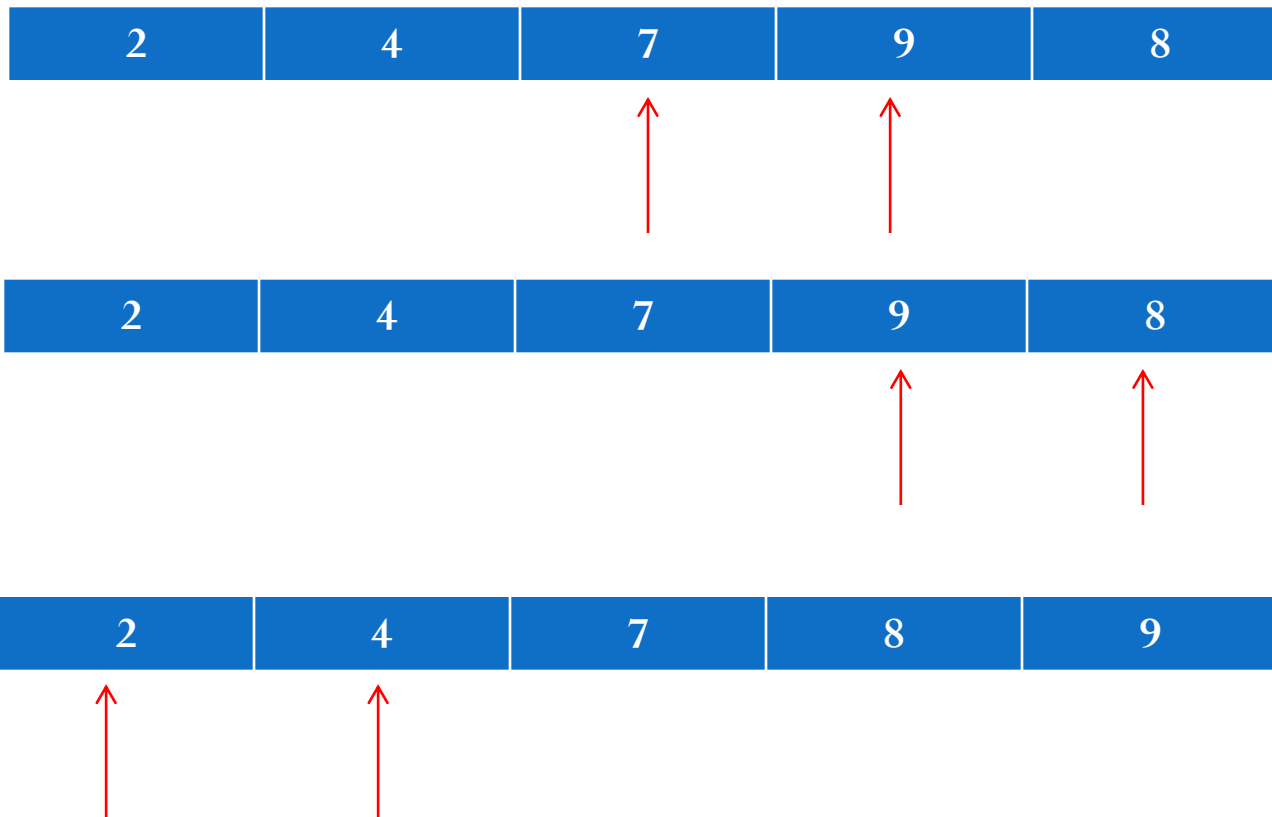
16

- Exemplo com o algoritmo melhorado



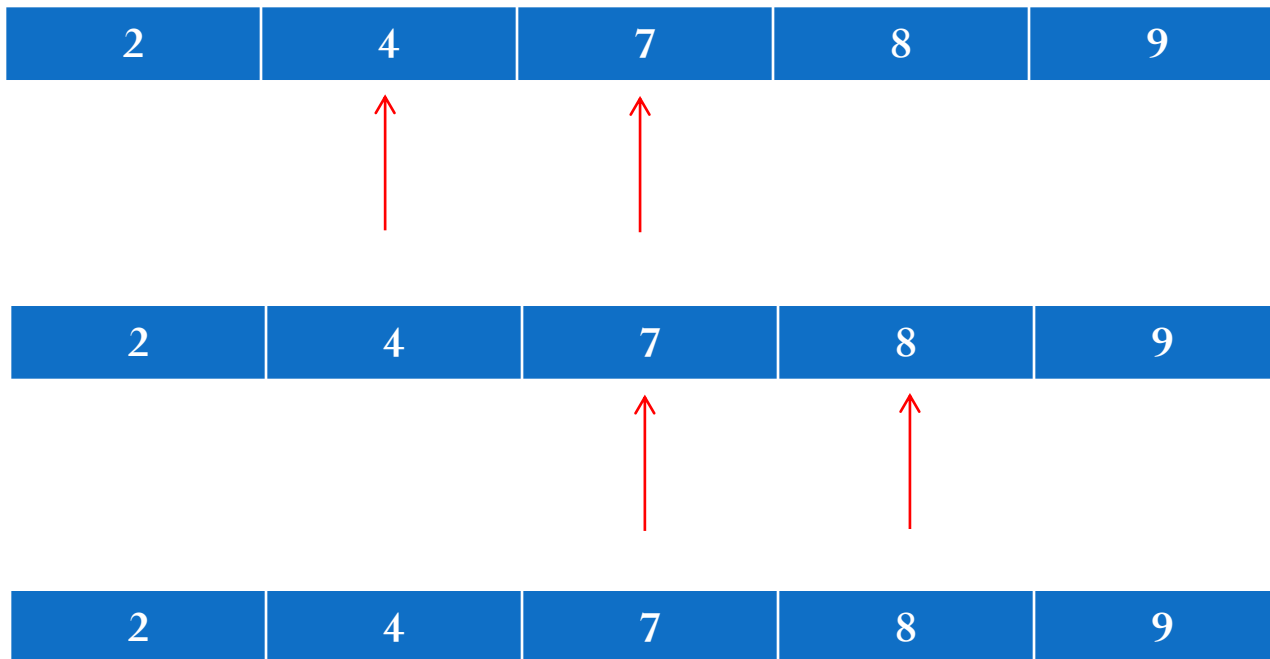
Bubble Sort

17



Bubble Sort

18



Bubble Sort

19

- Análise
 - Pior caso
 - Vetor na forma inversa
 - {5, 4, 3, 2}
 - Melhor caso
 - Vetor ordenado
 - {1, 2, 3, 4}
 - Número alto de comparações
 - Muito ruim na prática!

Bubble Sort

20

- Algoritmo rodando com o código
- <http://www.cs.oswego.edu/~mohammad/classes/csc241/samples/sort/Sort2-E.html>

Exercícios

21

- Laboratório 01
 - Implemente uma rotina para ordenar um vetor de 10 posições utilizando a técnica *Bubble Sort*.