



# Processamento da Informação

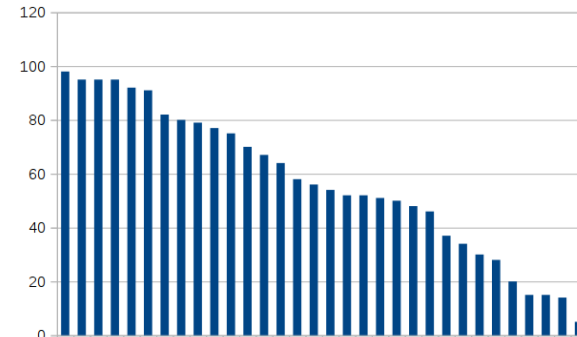
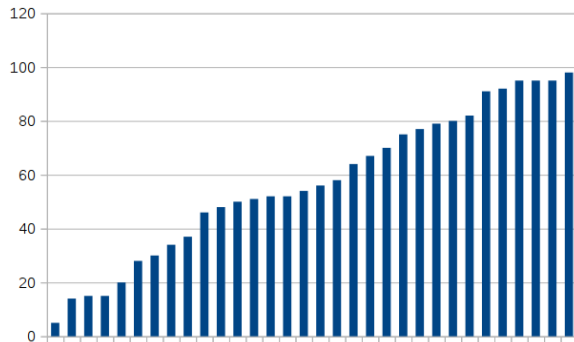
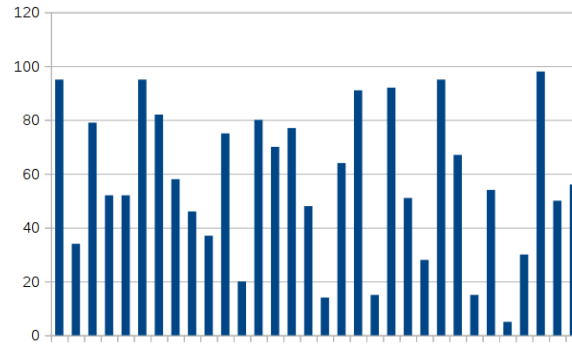
## Exercícios sobre vetores

Prof. Jesús P. Mena-Chalco  
CMCC/UFABC

Q2/2018

# Ordenando elementos

**Ordenar** corresponde ao processo de **re-arranjar (permutar) um conjunto de elementos** em ordem crescente ou decrescente.



# O problema de ordenar na forma crescente

Um vetor  $v[0..n-1]$  é crescente se  $v[0] \leq v[1] \leq \dots \leq v[n-1]$

## Vetores crescentes:

- $\{1, 1, 1, 1, 1, 1, 1, 1\}$
- $\{0, 1, 1, 1, 2, 3, 4, 4, 4, 4, 4, 4, 100\}$



**Verificar se um vetor  $v[0..n-1]$  é crescente**

```
import java.util.Scanner;
```

```
class Main
```

```
{  
    static boolean verificar_crescente(int v[]) {  
        int i;  
  
        for (i=0; i<v.length-1; i=i+1){  
            if (v[i]>v[i+1]) {  
                return false;  
            }  
        }  
        return true;  
    }  
}
```

```
public static void main(String[] args) {  
    int i;  
  
    Scanner entrada = new Scanner(System.in);  
    int n = entrada.nextInt();  
    int v[] = new int[n];  
  
    for (i=0; i<n; i=i+1){  
        v[i] = entrada.nextInt();  
    }  
  
    System.out.println( verificar_crescente(v) );  
}
```

```
}
```

```
> run Main
```

```
4
```

```
44 55 66 99
```

```
true
```

```
> run Main
```

```
7
```

```
33 44 55 66 77 88 0
```

```
false
```

```
>
```

```
import java.util.Scanner;
```

```
class Main
```

```
{
```

```
    static boolean verificar_crescente2(int v[]) {  
        int i=0;  
  
        while (i<v.length-1 && v[i]<=v[i+1]){  
            i = i+1;  
        }  
  
        return i==v.length-1;  
    }
```

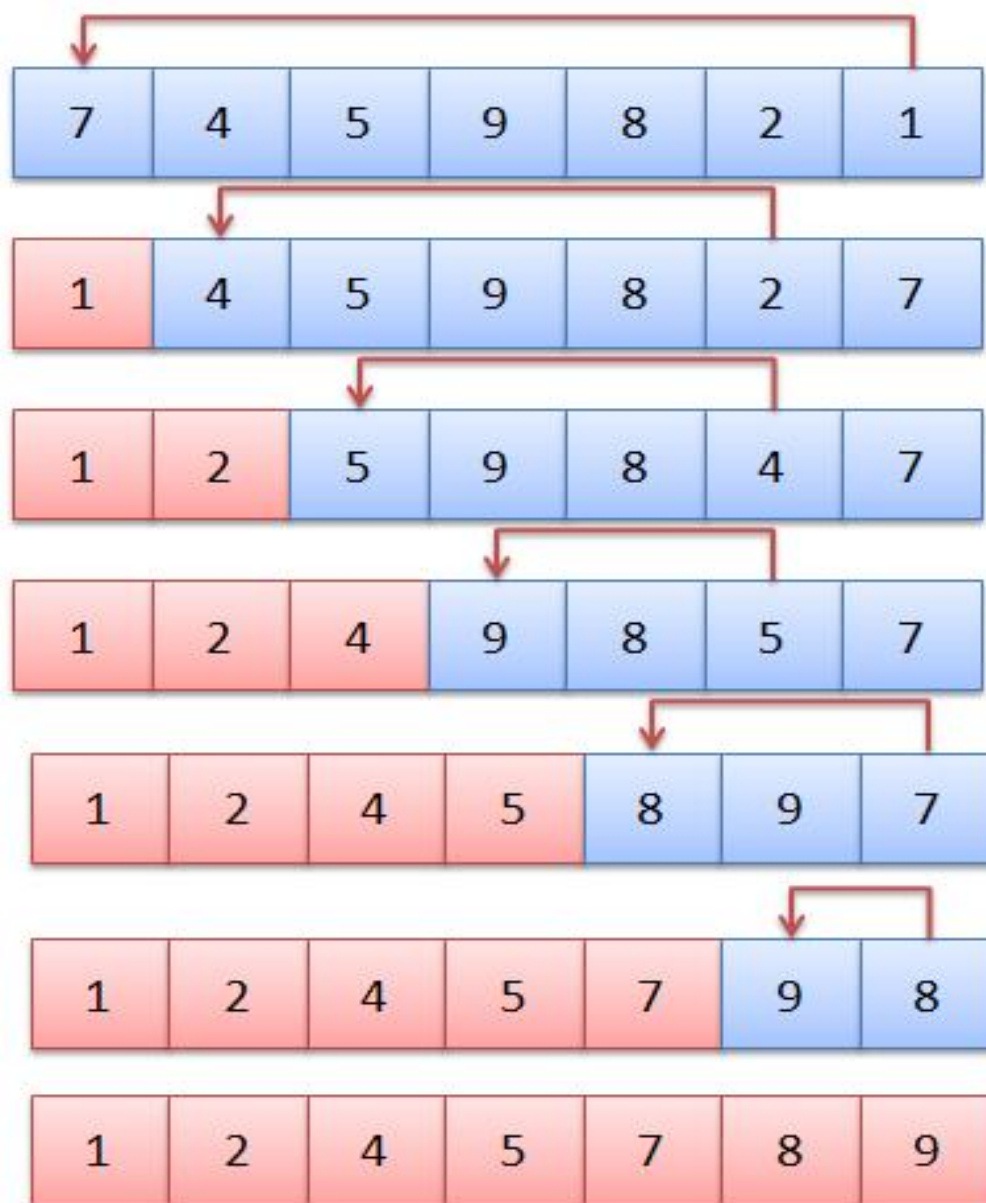
```
    public static void main(String[] args) {  
        int i;  
  
        Scanner entrada = new Scanner(System.in);  
        int n = entrada.nextInt();  
        int v[] = new int[n];  
  
        for (i=0; i<n; i=i+1){  
            v[i] = entrada.nextInt();  
        }  
  
        System.out.println( verificar_crescente2(v) );  
    }
```

```
}
```



# Ordenando números

# Selection Sort (ordenação por seleção)





```
import java.util.Scanner;
```

```
class Main
```

```
{  
    static void selection_sort(int v[]) {  
        int i, j, iMin, aux;  
  
        for (i=0; i<v.length-1; i=i+1) {  
            iMin = i;  
  
            for (j=i+1; j<v.length; j=j+1) {  
                if (v[iMin] > v[j]) {  
                    iMin = j;  
                }  
            }  
  
            aux      = v[iMin];  
            v[iMin] = v[i];  
            v[i]     = aux;  
        }  
    }  
}
```

```
static void imprimir_vetor(int v[]){  
    for (int k=0; k<v.length; k=k+1)  
        System.out.printf("%d ", v[k]);  
    System.out.printf("\n");  
}
```

```
public static void main(String[] args) {  
    Scanner entrada = new Scanner(System.in);  
    int n = entrada.nextInt();  
    int v[] = new int[n];  
  
    for (int i=0; i<n; i=i+1)  
        v[i] = entrada.nextInt();  
  
    imprimir_vetor(v);  
    selection_sort(v);  
    imprimir_vetor(v);  
}
```

```
}
```

```
> run Main
```

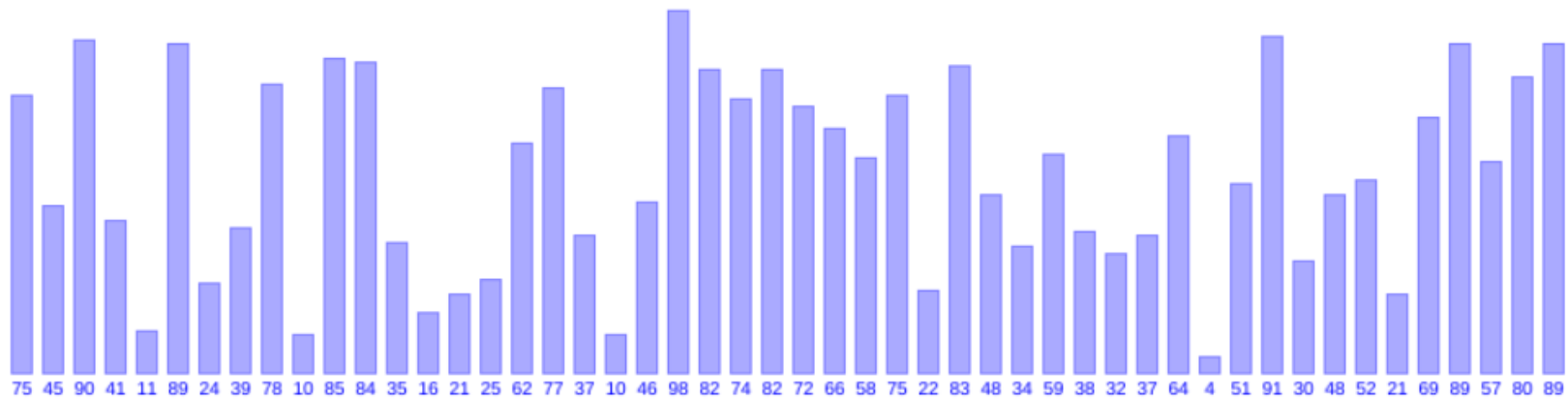
```
5
```

```
66 33 77 22 44
```

```
66 33 77 22 44
```

```
22 33 44 66 77
```

# Bubble sort



```
import java.util.Scanner;
```

```
class Main
```

```
{  
    static void bubble_sort(int v[]) {  
        int i, k, aux;  
  
        for (k=v.length-1; k>=1; k=k-1) {  
            for (i=0; i<k; i=i+1) {  
                if (v[i] > v[i+1]) {  
                    aux = v[i];  
                    v[i] = v[i+1];  
                    v[i+1] = aux;  
                }  
            }  
        }  
    }  
  
    static void imprimir_vector(int v[]){  
        for (int k=0; k<v.length; k=k+1)  
            System.out.printf("%d ", v[k]);  
        System.out.printf("\n");  
    }  
  
    public static void main(String[] args) {  
        Scanner entrada = new Scanner(System.in);  
        int n = entrada.nextInt();  
        int v[] = new int[n];  
  
        for (int i=0; i<n; i=i+1)  
            v[i] = entrada.nextInt();  
  
        imprimir_vector(v);  
        bubble_sort(v);  
        imprimir_vector(v);  
    }  
}
```

