

# Apresentação Professor

# Apresentação Professor

**Ricardo Molinari dos Prazeres**

# Formação Acadêmica

- **Graduação:** Processamento de dados – **FATEC**
- **Especialização:** Análise e desenv. de sistemas – **UFABC**
- **Licenciatura:** Centro Paula Souza
- **Mestrado:** Sistemas de Informação - **USP**

# Experiência Profissional

- **Estagiário:** Help desk
- **Programador:** Cobol / Java
- **Analista:** Sistemas bancários
- **Professor:** Desenv. de sistemas em geral

# Disciplina

## ESTRUTURA DE DADOS – 80 AULAS



# EMENTA

- Alocação dinâmica e ponteiros;
- Arquivos;
- Introdução à notação assintótica;
- Tipos abstratos de dados:  
conceitos, operações, representações, manipulação, listas, pilhas e filas.

# EMENTA

- Estruturas de representação de grafos (matriz de adjacência e de incidência);
- Estruturas para representação de árvores;
- Árvores binárias e suas aplicações.

# Alocação dinâmica e ponteiros



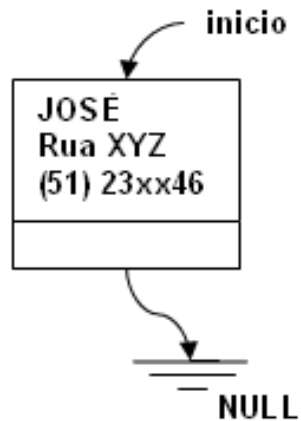
# Alocação dinâmica e ponteiros

## Array Estático

|     |      |      |      |     |       |
|-----|------|------|------|-----|-------|
| Ana | João | null | null | ... | null  |
| 0   | 1    |      |      |     | n - 1 |

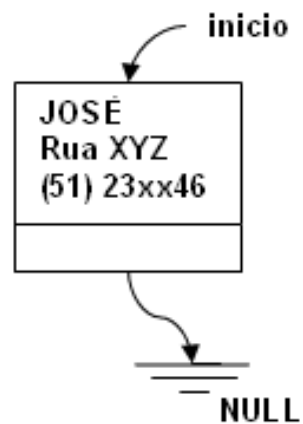
# Alocação dinâmica e ponteiros

## 1º Passo

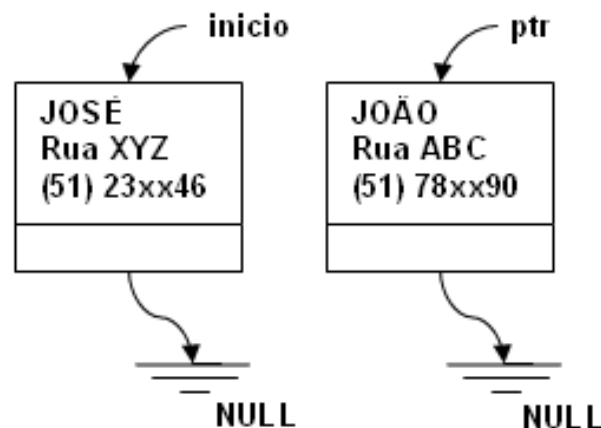


# Alocação dinâmica e ponteiros

## 1º Passo

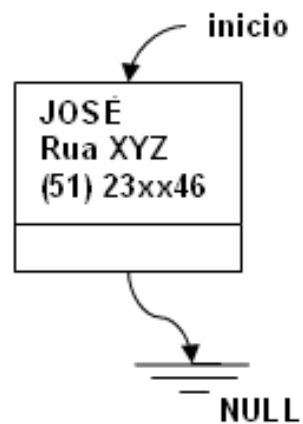


## 2º Passo

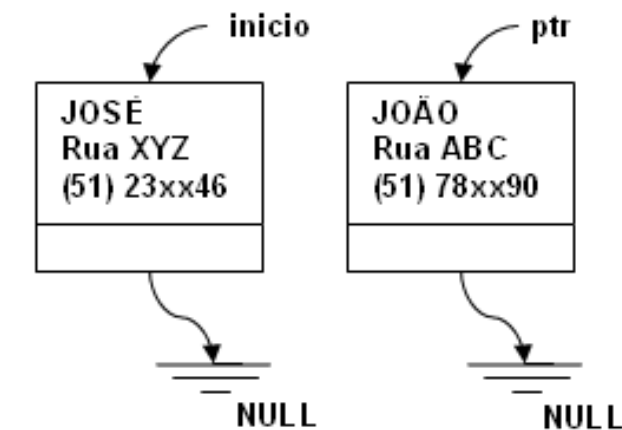


# Alocação dinâmica e ponteiros

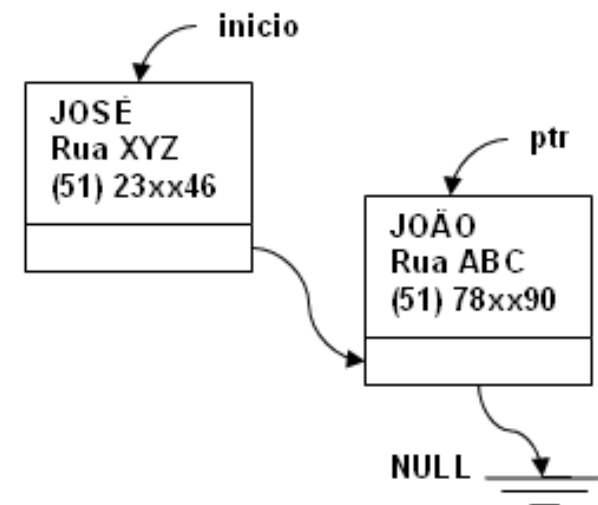
## 1º Passo



## 2º Passo



## 3º Passo



# Arquivos

# Arquivos

⚠ Não seguro | igm.univ-mlv.fr/AlgoB/gramofone/

## GraMoFoNe, a Plugin for Cytoscape

**GraMoFoNe**

**Download**

**Help & Snapshots**

**Publication**

**Contact**

### What is GraMoFoNe

GraMoFoNe (Graph Motif For Networks) is under [GPL licence](#), written in [Java 1.6](#) and developed as a plugin for [Cytoscape](#). Cytoscape is an open-source software platform for network visualization and analysis, which supports the development of external plugin tools extending its functionality.

Our plugin seeks for connected occurrences of a motif (set or multiset of proteins), given by the user, into a Protein-Protein Interaction network previously loaded into the Cytoscape workspace (many file format are supported). Similarity is measured among sequence-similarity via a [Blastp](#) analysis: proteins in the motif got different colors, proteins of the network are colored with the colors of motif proteins which are homologous. Note that a network node can have 0, 1 or more colors. GraMoFoNe allows deletions (proteins in the motif but not in the resulting connected subnetwork), and insertions (proteins in the results but not in the motif) of colored or uncolored network nodes.

It uses an exact algorithm to perform this task. To this end, we choose to express our problem as a linear pseudo-boolean optimization problem (LPB), *i.e.* as a linear program whose variables are boolean. We choose [Sat4JPseudo](#).

### Download

#### GraMoFoNe Program

- [Cytoscape](#) (has to be installed to run the plugin)
- [Blast](#) (has to be installed to run the plugin)
- GraMoFoNe Version 0.1 (2009-10)
  - The easiest way to install GraMoFoNe is to use the "Manage Plugin" menu in Cytoscape software
  - [Plugin Jar](#). Recommended. Save this file in the Cytoscape's Plugin directory.
  - [Source Code](#). If you want to modify GraMoFoNe. (The plugin (sources, javadoc, external libraries) and dirty program for batch tests)

#### Biological Data

- PPI Networks (from Torque website) as .sif files (can be imported in Cytoscape). Date : 2009-08.
  - [Saccharomyces cerevisiae](#) (Yeast, about 5.500 proteins and 40.000 interactions)
  - [Drosophila melanogaster](#) (Fly, about 6.500 proteins and 21.000 interactions)
  - [Homo sapiens](#) (about 8.000 proteins and 29.000 interactions)
- Motifs (kindly supplied by Torque authors) as .txt files. One motif by line, tab separated, with

# Arquivos



HomoSapiens\_Torque - Bloco de Notas

| Arquivo | Editar            | Formatar | Exibir | Ajuda |
|---------|-------------------|----------|--------|-------|
| 143379  | 0.412423346186800 |          |        | 56159 |
| 22903   | 0.182771084534359 |          |        | 5365  |
| 79098   | 0.412423346186800 |          |        | 9456  |
| 57586   | 0.182771084534359 |          |        | 9378  |
| 122769  | 0.182771084534359 |          |        | 3604  |
| 4681    | 0.182771084534359 |          |        | 7917  |
| 79363   | 0.412423346186800 |          |        | 80199 |
| 23564   | 0.412423346186800 |          |        | 9319  |
| 30819   | 0.182771084534359 |          |        | 3751  |
| 10742   | 0.412423346186800 |          |        | 1488  |
| 2978    | 0.182771084534359 |          |        | 3000  |
| 23132   | 0.182771084534359 |          |        | 367   |
| 11275   | 0.412423346186800 |          |        | 59349 |
| 27330   | 0.113033111253752 |          |        | 3164  |
| 4520    | 0.412423346186800 |          |        | 8907  |
| 154075  | 0.412423346186800 |          |        | 2189  |
| 28966   | 0.113033111253752 |          |        | 6457  |
| 2592    | 0.412423346186800 |          |        | 9319  |
| 27123   | 0.733226324163718 |          |        | 4040  |
| 11103   | 0.412423346186800 |          |        | 2130  |
| 10421   | 0.733226324163718 |          |        | 914   |
| 10421   | 0.182771084534359 |          |        | 24148 |
| 10725   | 0.182771084534359 |          |        | 50943 |
| 3207    | 0.182771084534359 |          |        | 51053 |
| 51450   | 0.182771084534359 |          |        | 6663  |

# Introdução à notação assintótica



# Introdução à notação assintótica

Foco em número **grande** de entrada de **dados**

# Introdução à notação assintótica

```
public boolean contem(Aluno aluno) {  
  
    for(int i = 0; i < totalDeAlunos; i++) {  
        if(aluno.equals(alunos[i])) {  
            return true;  
        }  
    }  
    return false;  
}
```

O que está fazendo?

# Introdução à notação assintótica

```
public boolean contem(Aluno aluno) {  
  
    for(int i = 0; i < totalDeAlunos; i++) {  
        if(aluno.equals(alunos[i])) {  
            return true;  
        }  
    }  
    return false;  
}
```

O que está fazendo?

# Introdução à notação assintótica

```
public boolean contem(Aluno aluno) {  
  
    for(int i = 0; i < totalDeAlunos; i++) {  
        if(aluno.equals(alunos[i])) {  
            return true;  
        }  
    }  
    return false;  
}
```

Quantas vezes será executado?

# Introdução à notação assintótica

$O$ ,  $\Omega$  e  $\Theta$

Pior, melhor e caso médio.

# Introdução à notação assintótica

## Classes O

|                         |             |
|-------------------------|-------------|
| $O(1)$                  | constante   |
| $O(\lg n)$              | logarítmica |
| $O(n)$                  | linear      |
| $O(n \lg n)$            | $n \log n$  |
| $O(n^2)$                | quadrática  |
| $O(n^3)$                | cúbica      |
| $O(n^k)$ com $k \geq 1$ | polinomial  |
| $O(2^n)$                | exponencial |
| $O(a^n)$ com $a > 1$    | exponencial |

# Introdução à notação assintótica

```
public boolean contem(Aluno aluno) {  
  
    for(int i = 0; i < totalDeAlunos; i++) {  
        if(aluno.equals(alunos[i])) {  
            return true;  
        }  
    }  
    return false;  
}
```

Quando é constante?

# Introdução à notação assintótica

```
public boolean contem(Aluno aluno) {  
  
    for(int i = 0; i < totalDeAlunos; i++) {  
        if(aluno.equals(alunos[i])) {  
            return true;  
        }  
    }  
    return false;  
}
```

Quando é polinomial?



# Introdução à notação assintótica



GPS sempre acha melhor rota?

# Tipos abstratos de dados:

Pilhas;

filas;

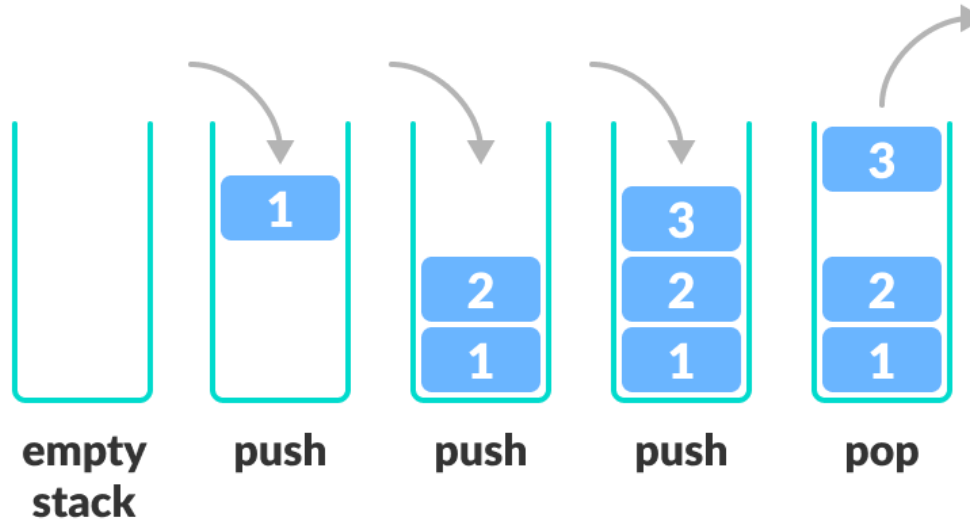
listas.

# Tipos abstratos de dados:

Pilhas

# Tipos abstratos de dados:

## Pilhas



# Tipos abstratos de dados:

## Pilhas

**LIFO:** Last In, First Out



# Tipos abstratos de dados:

## Pilhas

### **Principais operações:**

- estaVazia;
- estaCheia;
- empilhar;
- desempilhar;
- topo;

# Tipos abstratos de dados:

## Pilhas

### Fato interessante

supondo que tamanho = 3

estaCheia: verdadeiro

empilhar:

|   |
|---|
| C |
| B |
| A |

Pilha

# Tipos abstratos de dados:

## Pilhas

### Fato interessante

supondo que tamanho = 3

estaCheia: verdadeiro

empilhar:



|   |
|---|
| C |
| B |
| A |

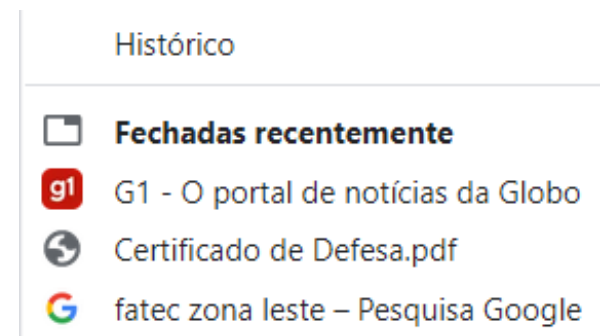
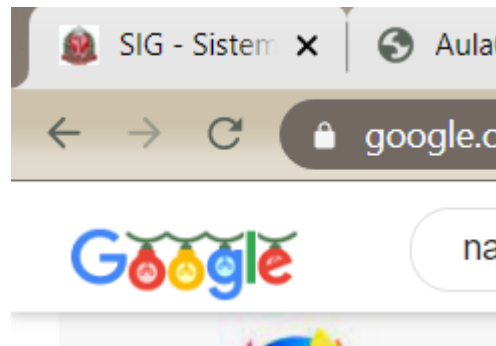
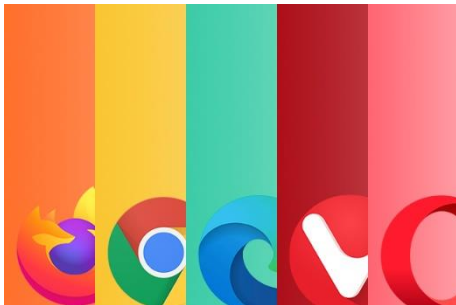
Pilha



# Tipos abstratos de dados:

Pilhas

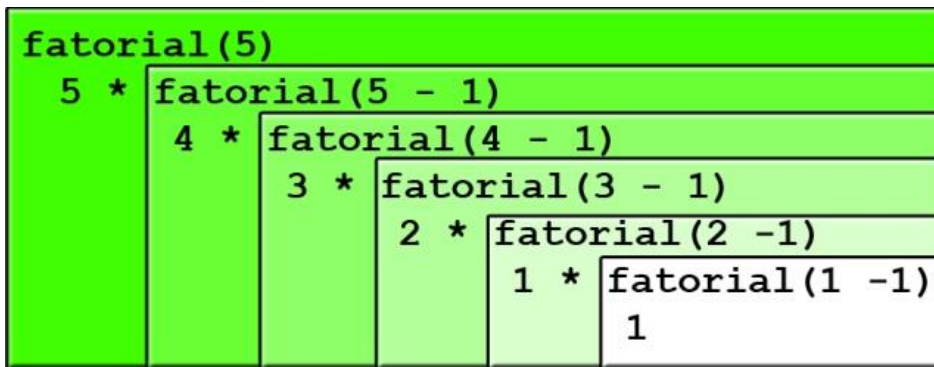
Histórico / navegação web



# Tipos abstratos de dados:

Pilhas

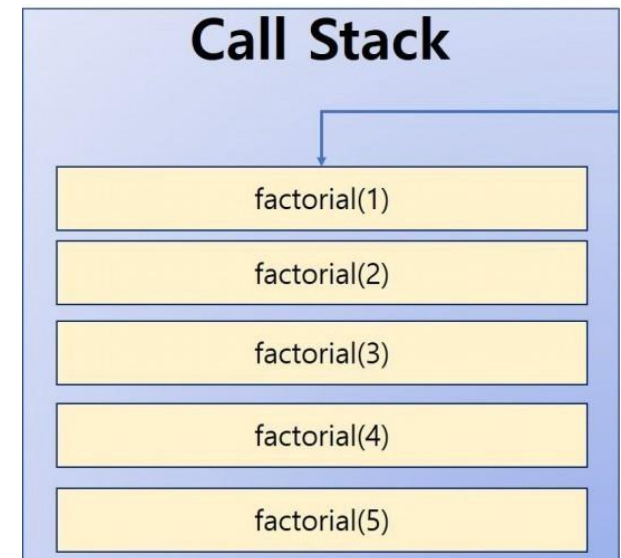
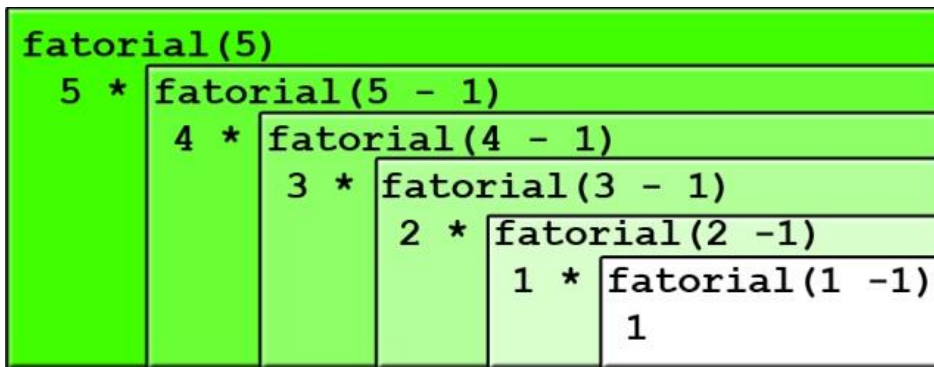
Recursividade



# Tipos abstratos de dados:

Pilhas

Recursividade



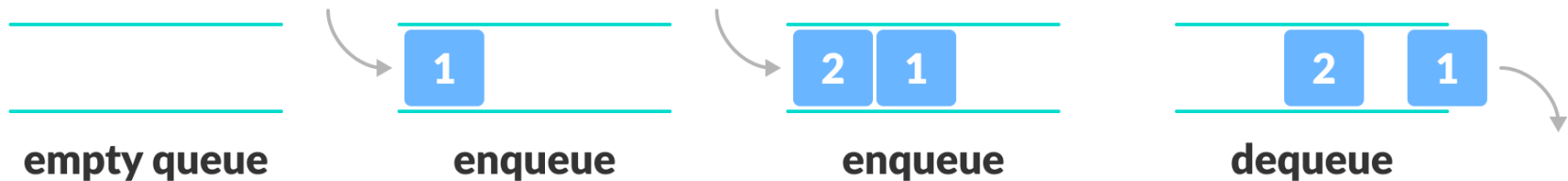
# Tipos abstratos de dados:

Filas

# Tipos abstratos de dados:

## Filas

**operações:** adiciona, remove, primeiro etc;

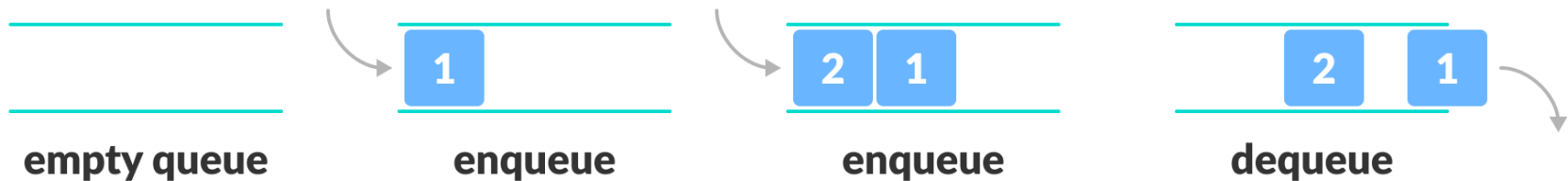


# Tipos abstratos de dados:

## Filas

**operações:** adiciona, remove, primeiro etc;

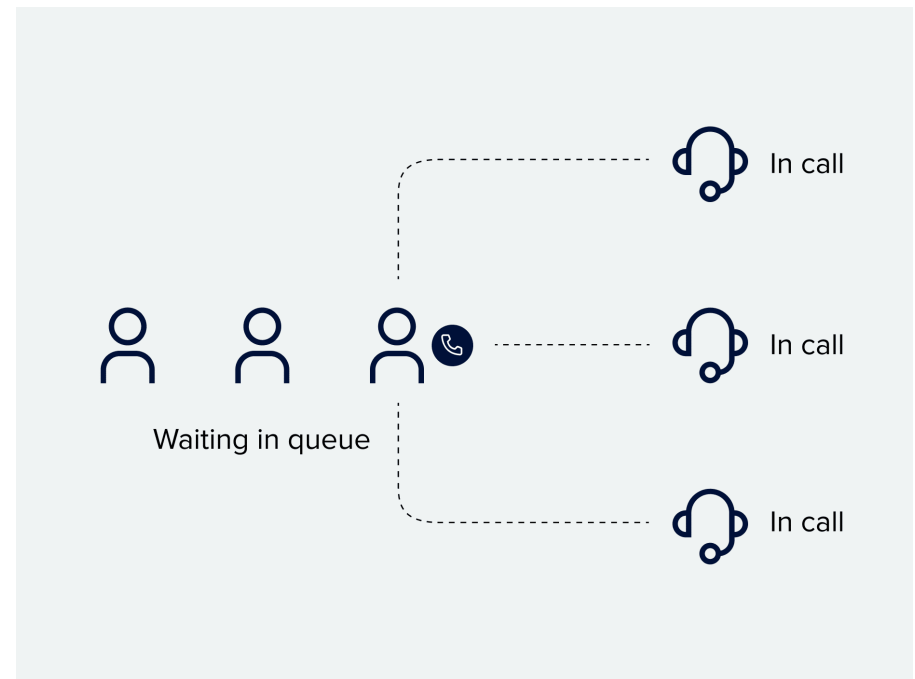
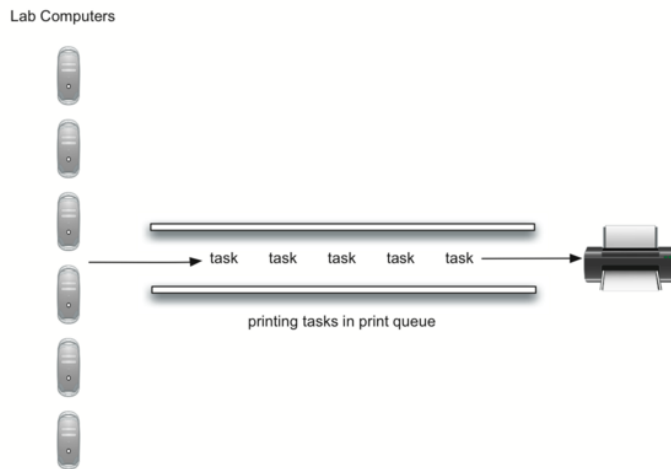
**FIFO:** First In, First Out



# Tipos abstratos de dados:

## Filas

### Aplicações:



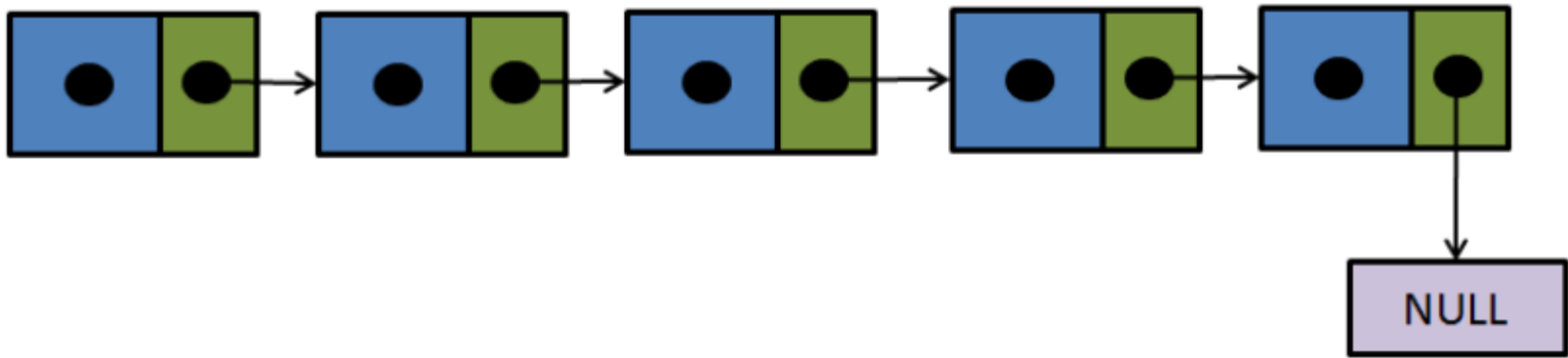
# Tipos abstratos de dados:

Listas



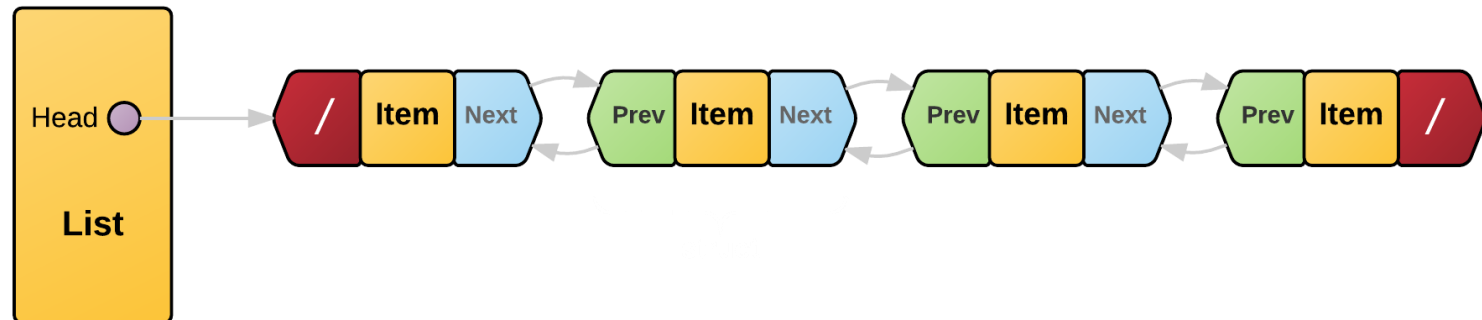
# Tipos abstratos de dados:

## Listas encadeadas



# Tipos abstratos de dados:

## Listas duplamente encadeadas

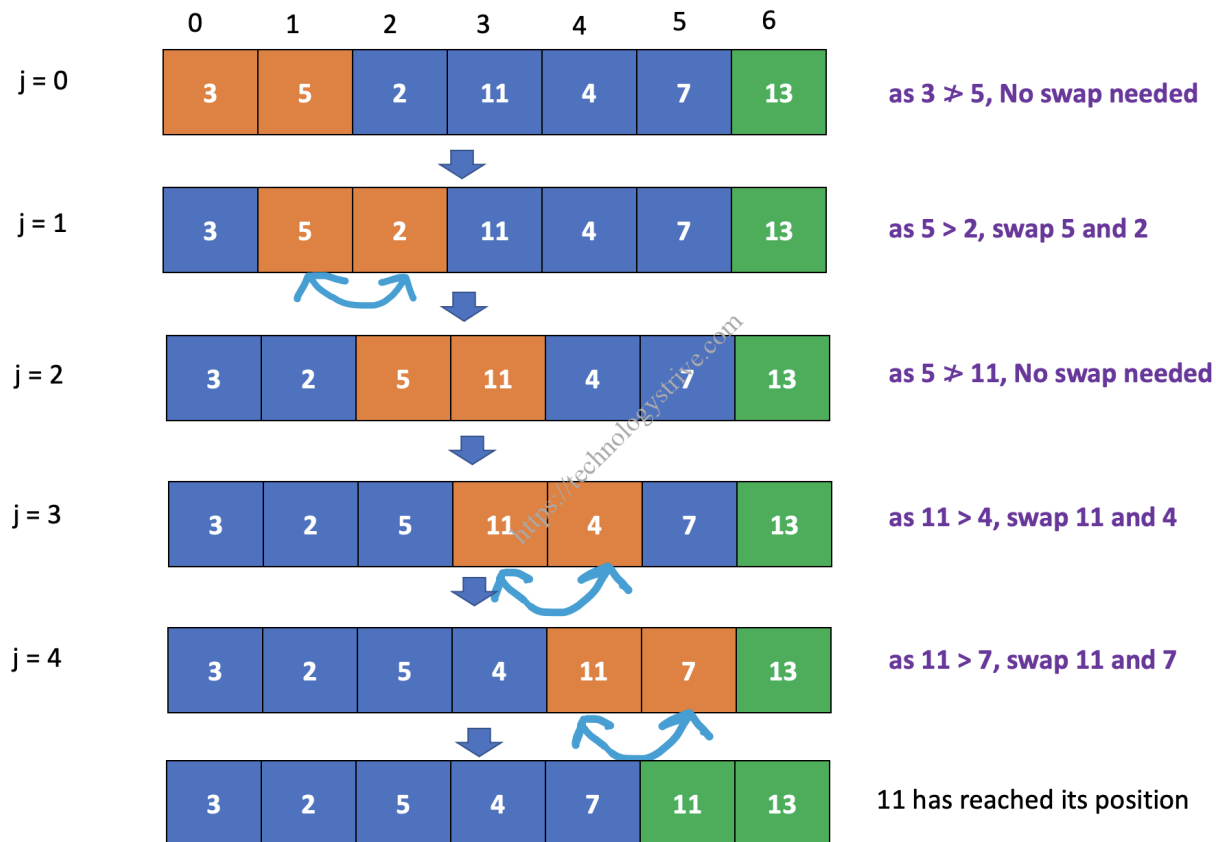


# Tipos abstratos de dados:

Ordenação (Sort)

# Tipos abstratos de dados:

## Ordenação (Sort): bubble



# Tipos abstratos de dados:

## Ordenação (Sort)

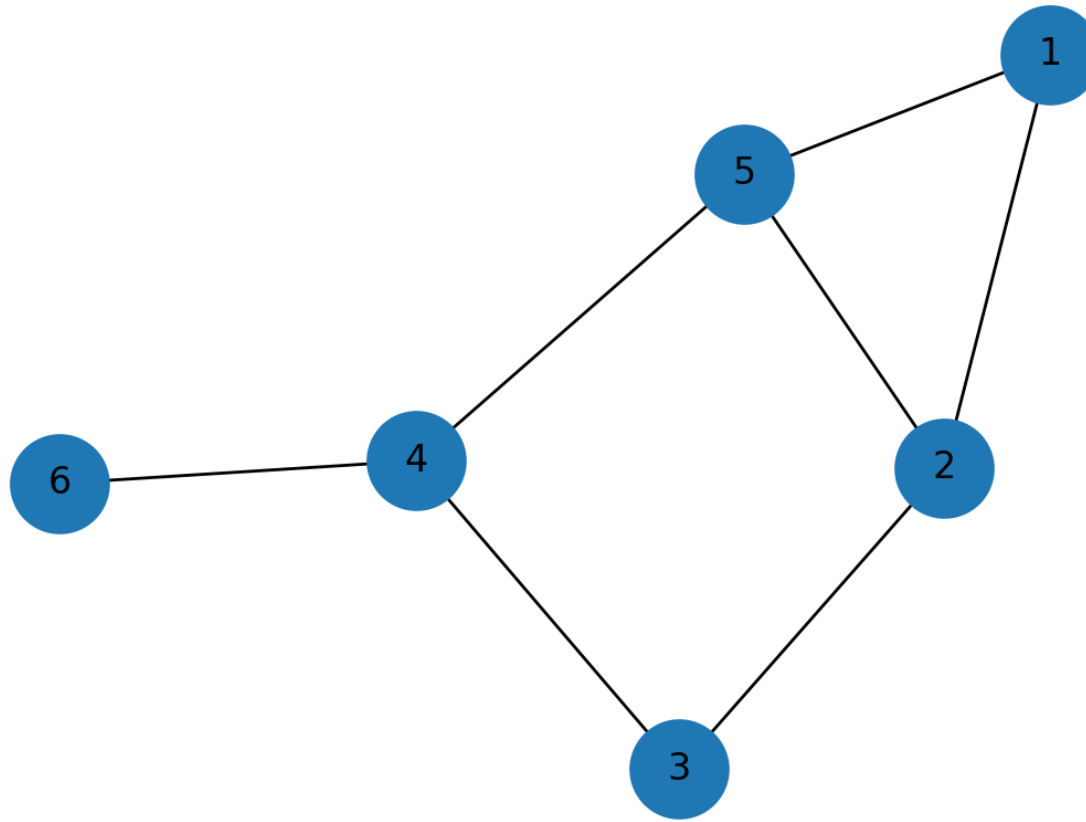
| Algoritmo      | Tempo           |                 |                 |
|----------------|-----------------|-----------------|-----------------|
|                | Melhor          | Médio           | Pior            |
| Merge sort     | $O(n \log_2 n)$ | $O(n \log_2 n)$ | $O(n \log_2 n)$ |
| Quick sort     | $O(n \log n)$   | $O(n \log n)$   | $O(n^2)$        |
| Bubble sort    | $O(n)$          | $O(n^2)$        | $O(n^2)$        |
| Insertion sort | $O(n)$          | $O(n^2)$        | $O(n^2)$        |
| Selection sort | $O(n^2)$        | $O(n^2)$        | $O(n^2)$        |

# Tipos abstratos de dados:

Grafos

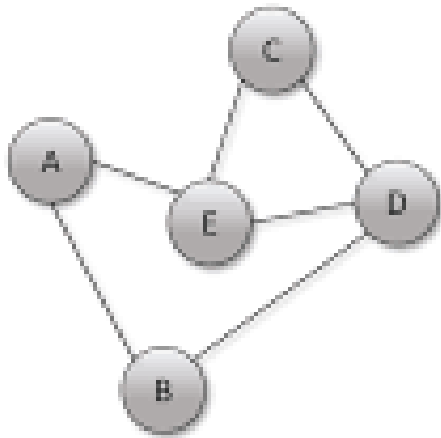
# Tipos abstratos de dados:

## Grafos

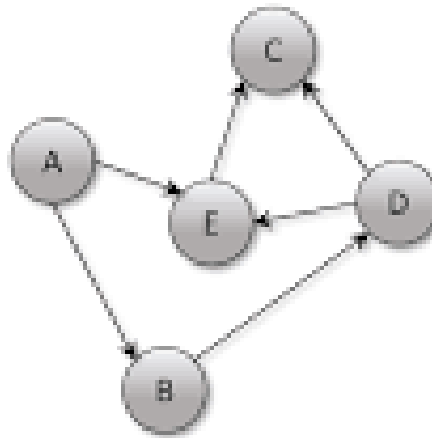


# Tipos abstratos de dados:

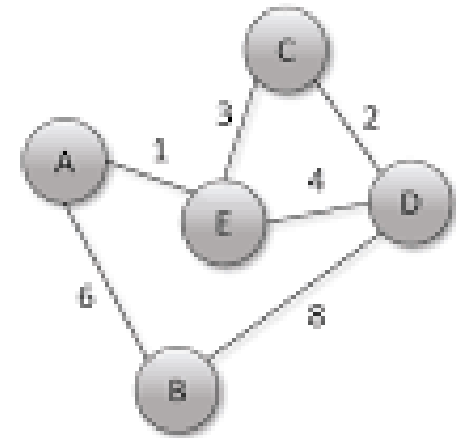
## Grafos



Não dirigido



Dirigido



Ponderado



# Tipos abstratos de dados:

## Grafos

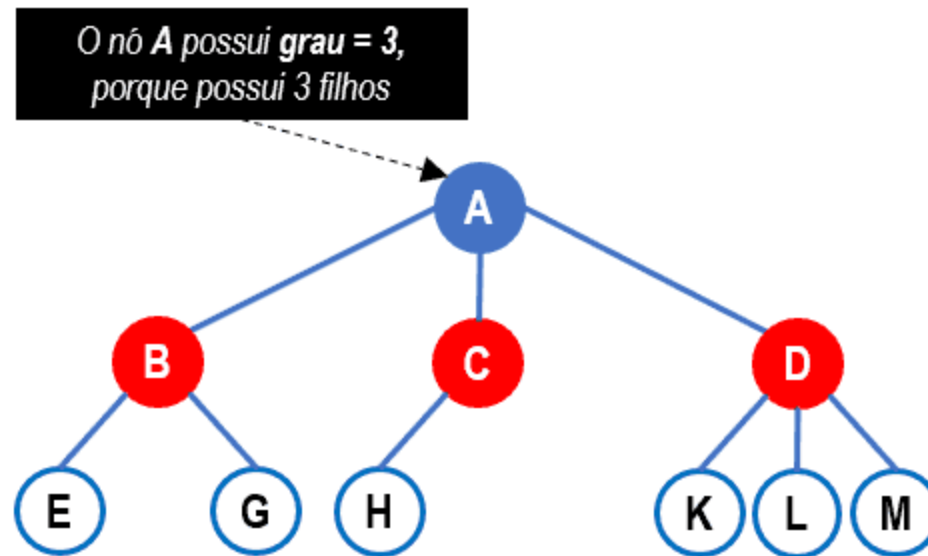


# Tipos abstratos de dados:

Árvores

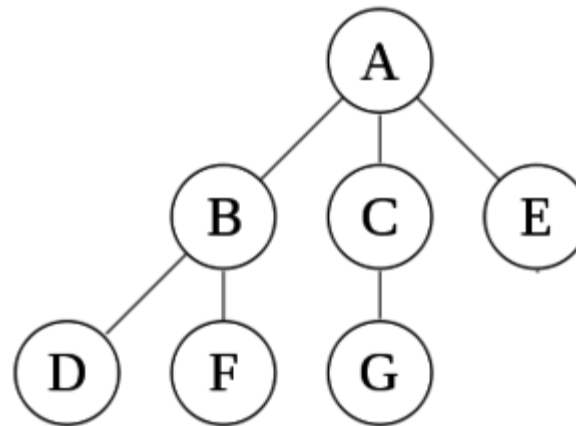
# Tipos abstratos de dados:

## Árvores



# Tipos abstratos de dados:

## Árvores binária



# MÉTODO

- Aulas expositivas
- Aulas práticas

# LINGUAGEM



# Bibliografia

GOODRICH, M. T.; TAMASSIA, R.  
Estruturas de Dados e Algoritmos em Java

CAELUM. CS-14 Algoritmos e Estruturas  
de Dados em Java

# Critérios Avaliativos

- Exercícios em laboratório
- Observação direta
- Avaliação individual (data?)

P1, P2 e PS



# Considerações Finais

# Revisão / Nivelamento