

## Instituto de Computação - Unicamp

MC102 - Algoritmos e Programação de Computadores

# Laboratório 19 - Hierarquia

---

Prazo de entrega: **28/06/2019 23:59:59**

Peso: **2**

*Professor:* Eduardo C. Xavier

*Professor:* Luiz Fernando Bittencourt

## Descrição

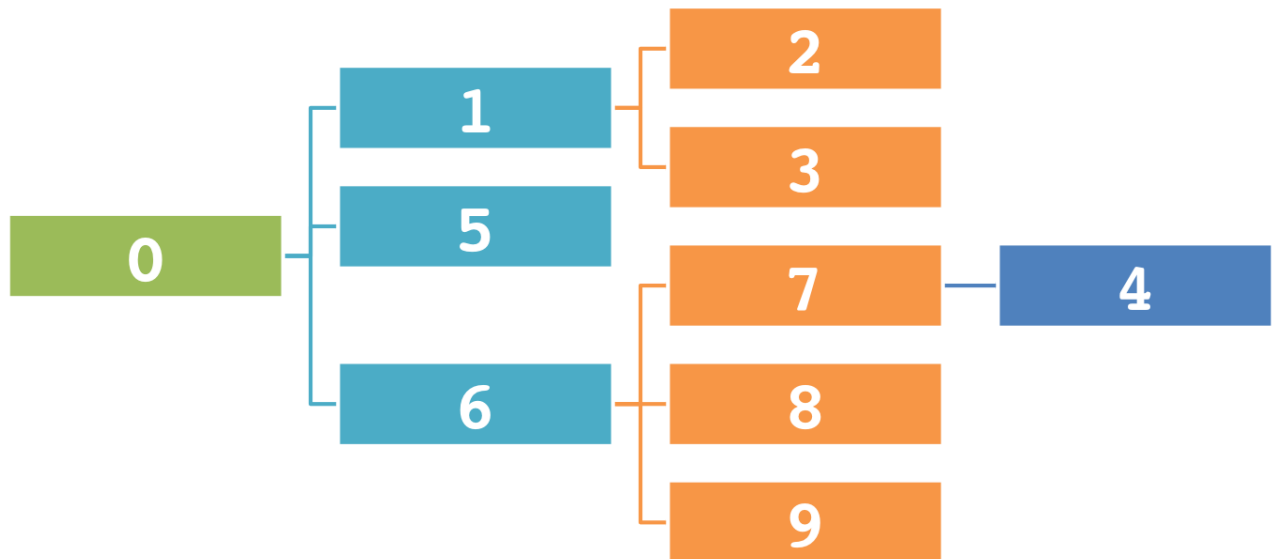
---

Esta tarefa tem por objetivo exercitar a construção de algoritmos recursivos.

Cargos em empresas geralmente são organizados de forma hierárquica. Isto permite que decisões que atingem apenas um (sub-)setor da empresa possam ser tomadas por líderes locais, não sobrecarregando os cargos mais altos na hierarquia.

A hierarquia da empresa pode ser entendida facilmente através de diagramas que mostram como ela está estruturada. Através deles, pode-se identificar a quem um determinado funcionário está diretamente subordinado, ou seja, quem é o seu *chefe*. Na verdade, pode-se até mesmo determinar toda a *cadeia hierárquica* de subordinados de um determinado funcionário, começando com o próprio funcionário e identificando-se quem são seus subordinados, em seguida quem são os subordinados dos seus subordinados e assim sucessivamente, até atingirmos os funcionários que não possuem subordinados.

Como veremos a seguir, o objetivo desta tarefa é encontrar a *cadeia hierárquica* de um funcionário de uma empresa. Antes porém, veja abaixo um exemplo de um desses diagramas hierárquicos onde cada funcionário é identificado por um número, sendo 0 (zero) o presidente da empresa.



Algumas relações que podemos ver no diagrama acima são:

- A cadeia hierárquica do funcionário 0 (o presidente) é formada pelos funcionários 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9 (todos os funcionários);
- A cadeia hierárquica do funcionário 1 é formada pelos funcionários 1, 2 e 3;
- A cadeia hierárquica do funcionário 7 é formada pelos funcionários 7 e 4;
- A cadeia hierárquica do funcionário 4 é formada somente por ele mesmo (pois não possui subordinados).

Podemos representar as informações do diagrama hierárquico de uma empresa com  $n$  funcionários fazendo uso de uma matriz quadrada de ordem  $n$ , com todos elementos tendo valor 0 (zero) ou 1 (um). Para tanto, vamos supor que os funcionários da empresa são identificados por números de 0 a  $n-1$ , sendo 0 (zero) o presidente da empresa. Assim, na linha  $i$  da matriz, uma coluna  $j$  terá o valor 1 (um) se o funcionário  $j$  for subordinado **direto** do funcionário  $i$ , caso contrário o valor nesta coluna é 0 (zero).

Por exemplo, a matriz correspondente ao diagrama hierárquico visto na imagem acima seria:

```

0 1 0 0 0 1 1 0 0 0
0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1
0 0 0 0 1 0 0 0 0 0

```

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

## Objetivo

---

O objetivo desta tarefa é fazer um programa que **use recursividade** e que, dada a matriz que descreve a hierarquia de uma empresa, encontre a *cadeia hierárquica* relativa a um determinado funcionário.

## Entrada

---

A primeira linha contém dois inteiros:  $n$ , o número de funcionários entre 3 e 30, e  $k$ , o identificador numérico do funcionário sobre o qual deseja-se conhecer a cadeia hierárquica. A seguir tem-se  $n$  linhas que correspondem as linhas da matriz que descrevem a hierarquia da empresa, conforme explicado no enunciado.

## Saída

---

Na saída devem ser impressos os números que identificam todos os funcionários que estejam na *cadeia hierárquica* do funcionário  $k$ , **começando pelo próprio**, e então imprimindo, **em ordem crescente** por identificador, os outros funcionários. Os números dos funcionários da cadeia devem estar separados por um único espaço em branco.

## Exemplos

---

### Teste 05

#### Entrada

```
5 0
0 1 1 0 0
0 0 0 1 1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

#### Saída

0 1 2 3 4

Teste 11

Entrada

10 6  
0 1 1 0 1 0 0 1 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 1 0 0 1 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 1  
0 0 0 0 0 0 0 0 1 0  
0 0 0 0 0 1 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0

Saída

6 5 8

Teste 10

Entrada

19 7  
0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0  
0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0  
0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0  
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

## Saída

7

Para mais exemplos, consulte os [testes abertos no Susy](#).

## Observações

---

- **O seu programa deve obrigatoriamente ser implementado de modo a usar recursividade.**
- O número máximo de submissões é **10**.
- O seu programa deve estar completamente contido em um único arquivo denominado `lab19.py`.
- Para a realização dos testes do SuSy, a execução do código em Python se dará da seguinte forma: (Linux e OSX) `python3 lab19.py`.
- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA.
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.

## CrITÉrios importantes

---

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará em nota zero nesta tarefa de laboratório.

- O aluno não poderá utilizar a função `sort` do python.