

Instituto de Computação da UNICAMP

Disciplina MC102: Algoritmos e Programação de Computadores - Turmas EF

Primeiro Semestre de 2015

Laboratório N° 18. Peso 2.

Prazo de entrega: **26/06/2015 às 23:59:59**

PROFESSOR: Alexandre Xavier Falcão

MONITORES: João do Monte Gomes Duarte

MONITORES: Jadisha Yarif Ramírez Cornejo

MONITORES: Takeo Akabane

MONITORES: Eduardo Spagnol Rossi

MONITORES: Guilherme Augusto Sakai Yoshike

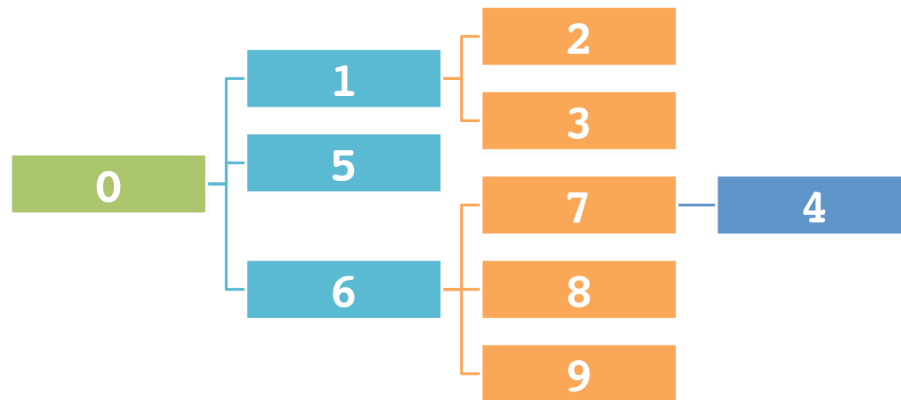
Hierarquia

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* abaixo da qual o funcionário está subordinado, começando com o próprio funcionário e identificando-se quem é o seu chefe, em seguida quem é chefe do seu chefe e, assim por diante, até chegar no presidente da empresa.

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 ZERO o presidente da empresa.



Algumas relações que podemos ver no diagrama acima são:

- A cadeia hierárquica do funcionário 4 é formada pelos funcionários 4, 7, 6 e 0;
- A cadeia hierárquica do funcionário 2 é formada pelos funcionários 2, 1 e 0;
- A cadeia hierárquica do funcionário 5 é formada pelos funcionários 5 e 0;
- A cadeia hierárquica do funcionário 0 é formada somente por ele mesmo (claro, pois ele é o presidente da empresa e, portanto, não é subordinado a ninguém).

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 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

```

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. Seguem abaixo os detalhes das especificações da entrada e da saída.

Entrada: uma linha contendo dois inteiros: N , o número funcionários entre 3 e 30, e K , o identificador numérico do funcionário sobre o qual deseja-se conhecer os superiores hierárquicos. A seguir tem-se N linhas que correspondem as linhas da matriz que descreve 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, até chegar ao presidente da empresa. Estes funcionários devem ser listados na ordem da menor para a maior hierarquia, ou seja, deve começar com **K** e terminar com 0 (o presidente). Os números dos funcionários da cadeia devem estar separados por um único espaço em branco.

Observações

- Você **deve** quebrar a linha após imprimir o resultado.
- O seu programa deve **obrigatoriamente** ser implementado de modo a usar recursividade.

Exemplos de execução

Exemplo 1:

```
4 0
0 1 1 0
0 0 0 0
0 0 0 1
0 0 0 0
0
```

Exemplo 2:

```
4 3
0 1 1 0
0 0 0 0
0 0 0 1
0 0 0 0
3 2 0
```

Exemplo 3:

```
10 2
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
```

2 1 0

Exemplo 4:

10 4

```
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 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
```

4 7 6 0

Notas: Textos em azul denotam dados de entrada do programa.

Textos em vermelho denotam dados de saída do programa.

Observações gerais: Observações

- O número máximo de submissões é 15;
- O seu programa deve estar completamente contido em um único arquivo denominado `hierarquia.c`;
- Para a realização dos testes do SuSy, a compilação se dará da seguinte forma: `gcc -std=c99 -pedantic -Wall -lm -o hierarquia hierarquia.c`;
- 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.

Critérios importantes

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

- O seu programa deve **obrigatoriamente** ser implementado de modo a usar recursividade.