



Pontifícia Universidade Católica de Minas Gerais  
Curso de Ciência da Computação - Coração Eucarístico  
Avaliação de Desempenho Acadêmico (ADA)

# Prova III - 10:00 às 12:50

Aluno: \_\_\_\_\_

## 1 CONTAR MAIÚSCULAS

Um problema trivial consiste em contar o número de letras maiúsculas ou minúsculas em uma frase.

**Dados de entrada:** Como entrada do programa, o arquivo texto contém uma frase a ser testada em cada linha. Ao encontrar a frase “FIM”, o teste deve ser finalizado.

**Exemplo de entrada:**

A cara rajada da jararaca  
A Cobra disse Cobre  
Quem muito Fala Nada Ouve  
Acorde, Pedroca  
FIM

**Dados de saída:** A saída produzida pelo programa é simples. Para cada frase testada, deve ser escrito em uma linha o número de letras maiúsculas contidas na mesma.

**Exemplo de saída:**

1  
3  
4  
2

## 2 PRÉ-ORDEM NAS ÁRVORES BINÁRIAS

Um das principais estruturas de dados são as árvores binárias. Nesse tipo de estrutura, quando a mesma está balanceada conseguimos efetuar as operações de inserção, remoção e pesquisa com  $O(\lg n)$  comparações.

**Dados de entrada:** Como entrada do programa, o arquivo texto contém várias linhas e, em cada uma delas, um conjunto de números sendo que o primeiro valor corresponde a quantidade de números na linha. A última linha contém apenas o número 0.

**Exemplo de entrada:**

```
9 1 2 3 4 5 6 7 8 9
9 9 8 7 6 5 4 3 2 1
9 5 3 8 1 2 4 6 7 9
9 9 7 6 4 2 1 8 5 3
3 1 2 3
3 3 2 1
0
```

**Dados de saída:** A saída produzida pelo programa é simples. Cada conjunto de entrada é inserido em uma árvore binária e sua pré-ordem (mostrar pré) é exibida.

**Exemplo de saída:**

```
1 2 3 4 5 6 7 8 9
9 8 7 6 5 4 3 2 1
5 3 1 2 4 8 6 7 9
9 7 6 4 2 1 3 5 8
1 2 3
3 2 1
```

### 3 UNICURSAL

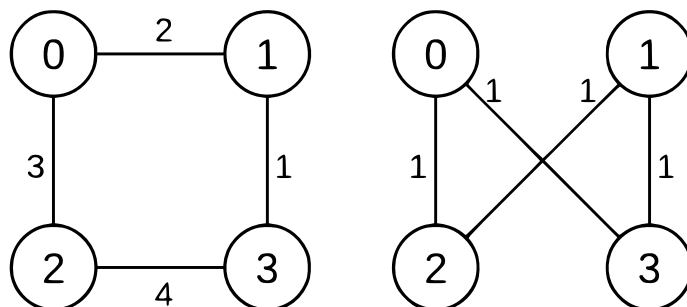
Um problema importante na teoria dos grafos é determinar se um grafo é unicursal.

**Dados de entrada:** A entrada padrão contém vários casos de teste sendo que cada um deles corresponde a um grafo. A primeira linha tem um número inteiro  $N$  indicando o número de vértices do primeiro grafo. Em seguida, temos  $N - 1$  linhas com as arestas do grafo. A primeira linha de arestas (segunda do arquivo) tem todas as arestas do primeiro vértice (zero). A linha seguinte tem todas as arestas do segundo vértice (um), contudo, nesse caso, não representamos aresta entre os dois primeiros vértices. A terceira linha de arestas não precisa representar as arestas entre o terceiro vértice e os dois primeiros. A  $N$ -ésima linha tem outro número inteiro e, se esse valor for zero, indica o término do arquivo. Caso contrário (como no exemplo abaixo), temos outro grafo.

**Exemplo de entrada:**

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

Este exemplo de entrada tem dois grafos (com quatro vértices cada) que estão representados nas figuras abaixo sendo que o primeiro grafo é o da esquerda.



**Dados de saída:** A saída produzida pelo programa é simples. Para cada grafo testado, deve ser escrito em uma linha do arquivo “SIM” caso o grafo seja unicursal, e “NAO” (sem acento) caso ela não seja.

**Exemplo de saída:**

```
NAO
NAO
```

## 4 BUSCA EM LARGURA

Um problema importante na teoria dos grafos é a busca em largura.

**Dados de entrada:** Igual a questão anterior.

**Dados de saída:** A saída produzida pelo programa é simples. Para cada grafo testado, deve ser escrito em uma linha do arquivo com a sequência de vértices a ser visitada. Quando um nó escolhe entre seus vizinhos aqueles a serem visitados, ele prioriza os vértices de menor índice. O vértice inicial é sempre o de menor índice (zero)

**Exemplo de saída:**

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