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

Prova III - 13:30 às 16:20 :

Aluno: _____

1 ENCONTRAR O MAIOR VALOR

Um problema importante na Ciência da Computação consiste em dado um conjunto de valores, encontrar o maior deles.

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:

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

Dados de saída: A saída produzida pelo programa é simples. Para cada conjunto de entrada, o maior valor é exibido.

Exemplo de saída:

```
7
9
7
9
3
2
```

2 LISTAS

As listas são um dos principais tipos de estruturas de dados onde podemos inserir elementos em qualquer posição da mesma. Podemos considerar que nossa lista tem sete operações básicas:

- **Inserir no início (II)** que insere um elemento na primeira posição de nossa estrutura.
- **Remover do início** que remove o primeiro elemento de nossa estrutura.
- **Inserir no fim (IF)** que insere um elemento no final de nossa estrutura (na verdade na primeira posição livre).
- **Remover do fim (IF)** que remove o último elemento de nossa estrutura.
- **Inserir qualquer (I)** que insere um elemento em uma posição qualquer (válida) recebida como parâmetro.
- **Remover qualquer (R)** que remove o elemento existente em uma posição qaulquer recebida como parâmetro.
- **Mostrar (M)** que mostra os elementos existentes em nossa estrutura.

Dados de entrada: Como entrada do programa, o arquivo texto contém várias linhas e, em cada uma delas, um comando a ser executado em nossa lista. A última linha contém apenas o número 0. Destaca-se que os comandos de inserção possuem um número inteiro indicando o valor a ser inserido na lista. O comando Inserir qualquer (I) tem dois inteiros, respectivamente, o valor a ser inserido e a posição de inserção. O comando de Remover qualquer (R) tem um inteiro indicando a posição de remoção. Finalmente, o comando de Mostrar (M) apenas exibe os elementos da lista na tela.

Exemplo de entrada:

```
II 1
IF 2
I 3 1
M
IF 4
IF 5
RI
RF
R 1
M
0
```

Dados de saída: A saída produzida pelo programa é simples. Corresponde as execuções do comando de mostrar.

Exemplo de saída:

```
1 3 2
3 4
```

3 REGULAR

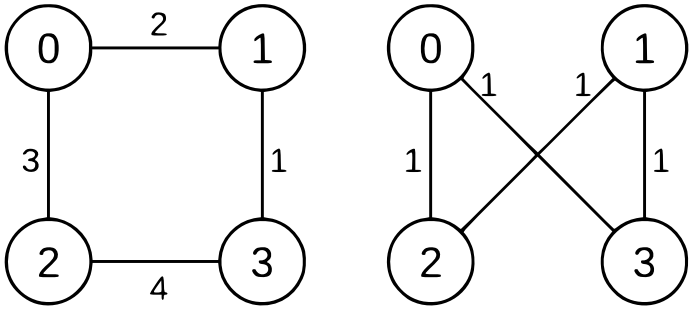
Um problema importante na teoria dos grafos é determinar se um grafo é regular.

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 regular, e "NAO" (sem acento) caso ela não seja.

Exemplo de saída:

```
SIM
SIM
```

4 BUSCA EM PROFUNDIDADE

Um problema importante na teoria dos grafos é o busca em profundidade.

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 indice. O vértice inicial é sempre o de menor indice (zero)

Exemplo de saída:

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