

Tarefa Nº 01 - Grafo - Listas de Adjacência

Prazo de entrega: **Consultar a página da tarefa.**

Linguagem para implementação: **C++**.

Professor: Andrei Braga

Grafo - Listas de Adjacência

Uma forma comum de representar computacionalmente um grafo é como *listas de adjacência*. Nesta tarefa, você deve implementar uma classe que use esta representação para armazenar um grafo **simples**. Você deve fazer isso utilizando **listas encadeadas** e escrevendo os seguintes métodos, que devem executar no grafo as operações indicadas:

- construtor Grafo: constrói o grafo com o número de vértices recebido por parâmetro e sem arestas;
- método `insere_aresta`: insere uma aresta no grafo de modo que
 - a aresta seja inserida caso ainda não exista no grafo e não seja um laço e
 - a alteração de uma lista de adjacência do grafo consista na inserção de um novo elemento no **início da lista**;
- método `remove_aresta`: remove uma aresta do grafo caso a aresta exista no grafo;
- método `num_arestas`: obtém o número de arestas do grafo;
- método `grau_maximo`: calcula o grau máximo do grafo;
- método `grau_minimo`: calcula o grau mínimo do grafo;
- método `imprime`: imprime o grafo conforme especificado na **Seção Saída** abaixo;
- (se necessário) destruidor `~Grafo`: libera a memória alocada para o grafo.

Você deve escrever um programa que constrói um grafo, executa operações no grafo e depois, se necessário, explicitamente o destrói. O seu programa deve processar informações que determinarão as operações a serem executadas no grafo, o que deve ser feito de acordo com as **Seções Entrada e Saída** abaixo.

Entrada

A primeira linha da entrada contém dois inteiros **V** ($V > 0$) e **O** ($O > 0$), sendo **V** o número de vértices do grafo a ser construído e **O** o número de operações a serem executadas no grafo. Cada uma das **O** linhas seguintes consiste em uma das opções abaixo:

- O caractere I, um inteiro **X** e um inteiro **Y** separados por espaços em branco;
- O caractere R, um inteiro **X** e um inteiro **Y** separados por espaços em branco;
- O caractere E;
- O caractere X;
- O caractere N;
- O caractere P.

Estas opções representam o seguinte, de acordo com o primeiro caractere da linha:

- Se é I, então deve ser executada a operação de inserir a aresta **X Y** no grafo (método `insere_aresta` – veja a **descrição acima**);
- Se é R, então deve ser executada a operação de remover a aresta **X Y** do grafo (método `remove_aresta` – veja a **descrição acima**);
- Se é E, então deve ser executada a operação de obter o número de arestas do grafo (método `num_arestas`);
- Se é X, então deve ser executada a operação de calcular o grau máximo do grafo (método `grau_maximo`);
- Se é N, então deve ser executada a operação de calcular o grau mínimo do grafo (método `grau_minimo`);
- Se é P, então deve ser executada a operação de imprimir o grafo (método `imprime`).

Saída

A saída deve consistir no seguinte:

- Para cada execução da operação de obter o número de arestas do grafo, o seu programa deve imprimir uma linha contendo o número de arestas obtido.
- Para cada execução da operação de calcular o grau máximo do grafo, o seu programa deve imprimir uma linha contendo o grau máximo calculado.
- Para cada execução da operação de calcular o grau mínimo do grafo, o seu programa deve imprimir uma linha contendo o grau mínimo calculado.
- Para cada execução da operação de imprimir o grafo, o seu programa deve imprimir **V** linhas, uma para cada um dos vértices 0, 1, ..., **V** - 1, em sequência. Cada uma destas linhas deve conter
 - o índice do vértice seguido do caractere : e
 - os índices dos vizinhos do vértice no grafo, cada um antecedido por um espaço em branco.

Os índices dos vizinhos do vértice devem ser impressos **na ordem** em que aparecem na sua **lista de adjacência**.

Exemplos de execução

Entrada	Saída
4 8 I 0 2 I 0 3 I 1 3 R 0 2 E X N P	2 2 0 0: 3 1: 3 2: 3: 1 0

Entrada	Saída
5 11 E X N R 1 2 I 1 0 R 1 0 I 1 3 E X N P	0 0 0 1 1 0 0: 1: 3 2: 3: 1 4:

Observações:

- Para a realização dos testes automáticos, a compilação se dará da seguinte forma:
`g++ -pedantic -Wall *.cpp -lm -lutil`
-