

Trabalho Nº 2 - Coloração

Prazo de entrega: **Consultar a página do trabalho.**

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

Professor: Andrei Braga

Coloração

Neste trabalho, você deve implementar uma classe que representa um **grafo simples** como uma **matriz de adjacências**. Você deve escrever os métodos a seguir, cujo objetivo é construir uma coloração dos vértices do grafo:

- `constroi_coloracao_1`
 - Corresponde ao algoritmo a seguir com os vértices sendo considerados em **ordem crescente dos seus índices**.
Algoritmo: descrito nas páginas 33 a 43 do material "Coloração de Grafos" disponibilizado no SIGAA na aula do dia 25/06/2025.
 - O método deve imprimir a coloração construída. Esta impressão deve consistir no seguinte:
 - Uma linha contendo o texto `Numero de cores :` seguido de um espaço em branco e do número de cores utilizado na coloração.
 - Para cada uma das **C** cores utilizadas na coloração (numeradas como 1, 2, ..., **C**), uma linha contendo
 - o texto `Cor` seguido de um espaço em branco, do número da cor e do caractere `:` e
 - os índices dos vértices coloridos com a cor.

Os índices dos vértices devem ser impressos **em ordem crescente**.

- `constroi_coloracao_2`
 - Corresponde ao algoritmo a seguir, onde os vértices são considerados em **ordem decrescente dos seus graus**.
Algoritmo: descrito na página 45 do material "Coloração de Grafos" disponibilizado no SIGAA na aula do dia 25/06/2025.
 - Em caso de vértices com o **mesmo grau**, devem ser considerados primeiro os vértices com menor índice.
 - O método deve imprimir a coloração construída. Esta impressão deve ser feita de maneira idêntica à descrita para o método `constroi_coloracao_1`.
 - **Dica:** Em C++, considere que `verts` é um `vector<int>` e que `metod` é um método que calcula um valor para um elemento de `verts`. Então, é possível ordenar os elementos de `verts` de maneira decrescente de acordo com o valor calculado por `metod` da seguinte maneira:

```
sort(verts.begin(), verts.end(), [&](int i, int j) { return metod(i) > metod(j); });
```


Para isto, é preciso adicionar a linha

```
#include <algorithm>
```


no início do arquivo onde está sendo feita a implementação.

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 **E** ($E \geq 0$), sendo **V** o número de vértices e **E** o número de arestas do grafo a ser construído. Cada uma das **E** linhas seguintes contém dois inteiros **X** e **Y**, indicando que o grafo contém uma aresta entre os vértices **X** e **Y**.

A próxima linha da entrada contém um caractere **M**, que representa o seguinte:

- Se é P, então deve ser executado apenas o método `constroi_coloracao_1` (descrito acima);
- Se é A, então devem ser executados ambos os métodos `constroi_coloracao_1` e `constroi_coloracao_2` (descritos acima);

Saída

A saída deve consistir no seguinte:

- Caso seja executado apenas o método `constroi_coloracao_1`, a impressão da coloração construída pelo método, como descrita acima.
- Caso sejam executados ambos os métodos `constroi_coloracao_1` e `constroi_coloracao_2`, a impressão da coloração construída pelo primeiro método seguida da impressão da coloração construída pelo segundo método, como descritas acima.

Exemplos de execução

Entrada	Saída
5 7 0 1 0 2 0 3 1 2 1 4 2 3 3 4 P	Numero de cores: 3 Cor 1: 0 4 Cor 2: 1 3 Cor 3: 2

Entrada	Saída
5 7 0 1 0 2 0 3 0 4 1 2 2 4 3 4 A	Numero de cores: 4 Cor 1: 0 Cor 2: 1 3 Cor 3: 2 Cor 4: 4 Numero de cores: 3 Cor 1: 0 Cor 2: 2 3 Cor 3: 1 4

Observações:

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