

Universidade Federal do Ceará - Campus Quixadá  
QXD0115 – Estrutura de Dados Avançada  
Curso de Ciência da Computação  
Prof. Atílio Gomes Luiz

Projeto Final
---------------

Entregue os exercícios de programação descritos neste documento até as 23h59 do dia **07/07/2023** pelo Moodle.

**Leia com cuidado as instruções abaixo.**

**Instruções:**

- Este trabalho deve ser feito **individualmente** ou **em dupla** e deve ser implementado usando a linguagem de programação C++
- Não se esqueça de colocar o **nome** e **matrícula** dos membros da equipe como comentário no seu código.
- Lembre-se de indentar corretamente o seu código para facilitar o entendimento.
- Os códigos-fonte devem estar devidamente organizados e documentados. Forneça bons comentários para cada função do seu programa, principalmente para aquelas que contiverem algoritmos não triviais.
- Obs.: Lembre-se de desalocar os endereços de memória alocados quando os mesmos não forem mais ser usados.
- Obs.: Qualquer indício de plágio resultará em nota **ZERO** para todos os envolvidos.

**DICA: COMECE O TRABALHO JÁ.**

### 1. [Problema 1] Colorindo os vértices do grafo com duas cores

Você já coloriu um mapa? Pegue um mapa qualquer dividido em regiões e uma caixa de lápis de cores. Para colorir este mapa sem que duas regiões vizinhas tenham a mesma cor, de quantas cores você vai precisar? Apenas quatro. Isso é o que diz o TEOREMA DAS QUATRO CORES.

Por volta de 1852, um estudante inglês, chamado **Francis Guthrie**, tentou colorir o mapa com os distritos da Inglaterra e, refletindo sobre o problema, conjecturou que com 4 cores seria possível pintar qualquer mapa sem que regiões vizinhas tivessem a mesma cor.



Figura 1: Mapa do Brasil colorido com 4 cores.

O problema se popularizou e passou pelas mãos de muitos matemáticos famosos. Mesmo com tamanho engajamento de tantas mentes inventivas, demorou mais de 100 anos até que o fato fosse demonstrado pela primeira vez em 1976 por Kenneth Appel e Wolfgang Haken, utilizando um computador IBM 360, que teve de realizar bilhões de cálculos. Mas essa polêmica não chegou ao fim: a demonstração obtida por Appel e Haken exige tantos cálculos que se torna humanamente impossível realizá-la sem auxílio de computadores.

O fato é que esse problema de coloração de mapas é equivalente ao problema de tentar colorir os vértices de um grafo planar de modo que quaisquer dois vértices vizinhos tenham cores distintas (Um grafo *planar* é um grafo que pode ser desenhado no plano



(a) Francis Guthrie



(b) Appel e Haken

Figura 2: Francis propôs a Conjetura das Quatro Cores e Appel e Haken a provaram.

de modo que nenhuma de suas arestas se cruzem). A modelagem do problema de coloração de mapas como um problema em grafos se dá da seguinte forma: cada região do mapa se torna um vértice do grafo e adicionamos uma aresta conectando dois vértices  $u$  e  $v$  no grafo se e somente se as regiões que eles representam no mapa são regiões vizinhas, ou seja, regiões que têm uma fronteira em comum. Daí, colorir regiões do mapa se traduz em colorir vértices do grafo. Por exemplo, a Figura 3 exibe um grafo planar com seus vértices coloridos de modo que quaisquer dois vértices adjacentes têm cores distintas.

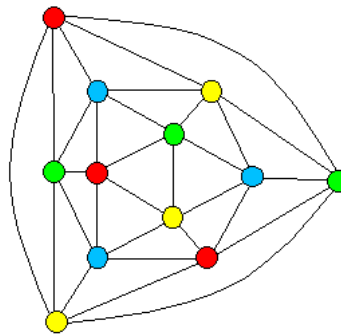


Figura 3: Um grafo planar com uma 4-coloração de vértices válida.

Uma **coloração dos vértices** de um grafo não-direcionado é uma atribuição de cores aos seus vértices. Dizemos que uma coloração de vértices é **válida** se quaisquer dois vértices vizinhos do grafo têm cores diferentes. Por exemplo, o grafo da Figura 3 é apresentado com uma coloração de vértices válida.

O PROBLEMA DA  $k$ -COLORAÇÃO DE VÉRTICES é o seguinte: Dado um grafo não-direcionado  $G$  e um inteiro  $k$ , encontrar uma coloração válida de  $G$  com  $k$  cores.

O problema da coloração de vértices de grafos possui muitas aplicações práticas como por exemplo, na determinação do menor número de registradores utilizados por compiladores. Até alcançou popularidade com o público em geral na forma do quebra-cabeças popular chamado [Sudoku](#).

## Trabalho

Para este problema, implemente um algoritmo em C++ que:

- Dado um grafo não direcionado  $G = (V, E)$  como entrada, retorne **false** se não existir uma coloração de vértices válida do grafo com **duas** cores; e retorne **true**, caso exista tal coloração.

## Entrada

O seu programa deverá ler os grafos a partir de um arquivo texto chamado `grafos.txt`. Haverá vários casos no arquivo de entrada. A primeira linha de cada caso contém o número de vértices,  $N$ , seguido do número  $M$  de arestas do grafo. As próxima linha de cada caso contém  $M$  pares de inteiros,  $a$  e  $b$ , que são os vértices extremos de uma aresta ( $0 \leq a, b \leq N - 1$ ).

## Saída

Para cada grafo lido na entrada, imprima na tela ‘SIM’ se o grafo tiver uma coloração de vértices válida com duas cores (as cores devem ser indicadas pelas letras R e B) e ‘NAO’ caso contrário.

## Exemplo de entrada

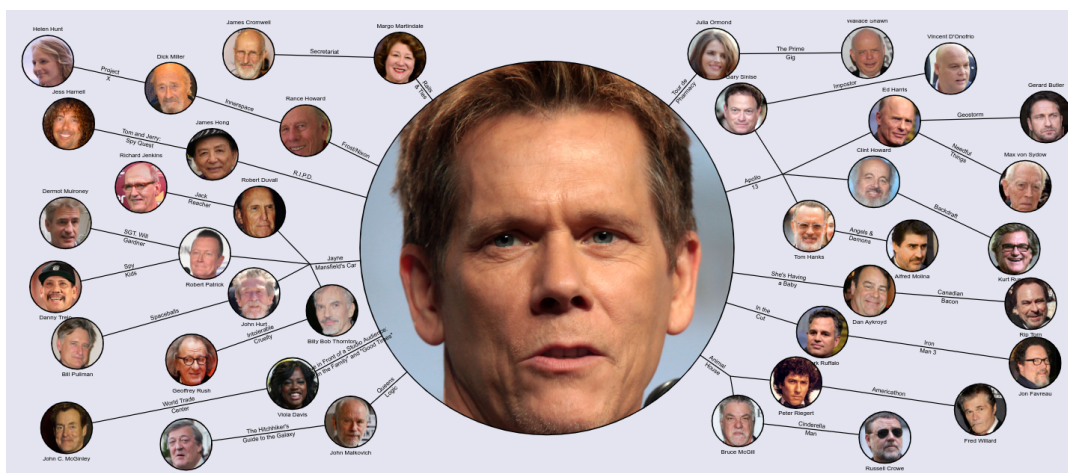
```
4 6
0 1 1 2 2 3 0 3 1 3 0 2
6 9
0 3 0 4 0 5 1 3 1 4 1 5 2 3 2 4 2 5
```

## Exemplo de saída

```
NAO
SIM
```

**Observação:** Nesta questão, represente grafos como lista de adjacências.

## 2. [Problema 2] Seis graus de Kevin Bacon



Uma aplicação divertida da teoria dos grafos surge no jogo [Seis graus de Kevin Bacon](#).

A ideia do jogo é conectar um ator a [Kevin Bacon](#) por meio de uma trilha de atores que apareceram juntos em filmes, e isso em menos de seis etapas. Por exemplo, Theodore Hesburgh (presidente emérito da Universidade de Notre Dame) esteve no filme [Rudy](#) com o ator Gerry Becker, que esteve no filme [Sleepers](#) com Kevin Bacon, então seu número de Bacon é 2. Claramente, o número de Bacon do próprio Kevin Bacon é 0.

Por que Kevin Bacon? Por algum motivo, os [três alunos](#) que inventaram o jogo em 1994, Mike Ginelli, Craig Fass e Brian Turtle, decidiram que Kevin Bacon é o centro do mundo do entretenimento. O jogo Kevin Bacon é bastante semelhante ao Número de Erdős, que faz parte do folclore dos matemáticos em todo o mundo há muitos anos.

O jogo Seis Graus de Kevin Bacon é realmente um problema de grafos. Se você atribuir cada ator a um vértice e adicionar uma aresta entre dois atores se eles aparecerem juntos em um filme, você terá um grafo que representa os dados para este jogo. Então, o problema de encontrar uma trilha de atores até Kevin Bacon torna-se um problema tradicional em grafos: o de encontrar um caminho entre dois vértices. Uma vez que desejamos encontrar um caminho mais curto do que seis etapas, o ideal seria encontrar o caminho mais curto entre os vértices. Pode-se pensar em aplicar o algoritmo de caminho mais curto de Dijkstra a esse problema, mas isso seria um exagero, pois o algoritmo de Dijkstra se destina a situações em que cada aresta tem um comprimento (ou peso) associado e o objetivo é encontrar o caminho com o comprimento cumulativo mais curto. Uma vez que estamos apenas preocupados em encontrar os caminhos mais curtos em termos do número de arestas, um dos algoritmos de busca em grafos irá resolver o problema (e usará menos tempo do que o de Dijkstra).

O jogo de Kevin Bacon é baseado na Teoria dos Seis Graus de Separação. Essa teoria diz que todas as pessoas estão interligadas por um número pequeno de conexões. A origem não é científica, mas literária: foi criada em 1929 pelo escritor húngaro Frigyes Karinthy no livro *Tudo É Diferente*. Para Karinthy, os avanços na comunicação e

nos transportes fariam com que, apesar das distâncias entre as pessoas, os círculos sociais ficassem cada vez menores. A partir dessa hipótese, um personagem especulou a ligação entre um operário da Ford, nos EUA, e ele, em Budapeste. A ficção inspirou a realidade, e a ideia serviu de base para estudos de matemática, computação e ciências sociais que abordam conceito de redes de influência e difusão de informação.

## Entrada

A entrada consiste em uma lista de pares de atores que apareceram no mesmo filme. Cada linha do arquivo contém o nome de um ator, um filme e outro ator que apareceu no filme. O caractere “ ; ” (ponto-e-vírgula) é usado como separador. Aqui está um trecho contendo 4 linhas de uma possível entrada:

```
Clint Howard;My Dog Skip (2000);Kevin Bacon  
Sean Astin;White Water Summer (1987);Kevin Bacon  
Theodore Hesburgh;Rudy (1993);Gerry Becker  
Gerry Becker;Sleepers (1996);Kevin Bacon
```

## Saída

Cada linha da saída do programa deve conter uma frase indicando o nome de um ator seguido pelo seu número de Bacon e indicando o filme pelo qual o ator ganha esse número. Um exemplo contendo 4 linhas de uma possível saída é exibido a seguir:

```
O numero de Bacon de Angelo Rossitto é 2 pelo filme Dark, The (1979)  
O numero de Bacon de Bebe Drake é 2 pelo filme Report to the Commissioner  
(1975)  
O numero de Bacon de Bill Paxton é 1 pelo filme Apollo 13 (1995)  
O numero de Bacon de Carrie Fisher é 2 pelo filme Soapdish (1991)
```

Os nomes devem aparecer em ordem alfabética.

**Observação:** Nesta questão, represente grafos como lista de adjacências.

### 3. [Problema 3] Vias de mão dupla



De acordo com o Departamento de Trânsito do estado do Ceará (DETRAN-CE), a maioria dos acidentes fatais de trânsito ocorre em vias de mão dupla.

Hoje em dia, o trânsito em [Pedra Branca](#), município do interior do estado do Ceará, é de mão dupla em todos os trechos de rua e é possível alcançar cada esquina a partir de qualquer outra. A fim de reduzir o número de mortes causadas por acidentes de trânsito, o prefeito de Pedra Branca, quer reorientar o trânsito e converter tantas ruas quanto possível em ruas de mão única.

Seguindo o mesmo exemplo do prefeito de Pedra Branca, vários outros prefeitos de outros municípios do interior do estado também querem converter as ruas de suas cidades para ruas de mão única.

Você foi contratado para realizar esta conversão de modo que, a partir de qualquer esquina na cidade, seja possível para um motorista dirigir para todas as outras esquinas seguindo alguma rota.

Como entrada do problema, você receberá uma lista das ruas de uma cidade (todas de mão dupla). Cada rua da cidade conecta duas esquinas, e não ultrapassa uma esquina, ou seja, ao chegar numa esquina, a rua termina. Além disso, não há mais do que uma rua conectando qualquer par de esquinas. É possível também ter ruas que comecem em uma esquina, mas que terminem sem saída, ou seja o outro extremo da rua é uma esquina que é um beco sem saída.





## Entrada

O seu programa deverá ler os grafos a partir de um arquivo texto chamado `grafos.txt`. A entrada consiste em vários casos. A primeira linha de cada caso contém dois inteiros  $n$  e  $m$ . O número de cruzamentos (esquinas) é  $n$  ( $2 \leq n \leq 1000$ ) e o número de ruas é  $m$ . A próxima linha é formada por  $m$  pares representando as esquinas de cada uma das  $m$  ruas (as extremidades dos vértices). Os vértices do grafo são numerados de 0 a  $n - 1$ . Além disso, cada rua é listada uma única vez, ou seja, se o par  $(i, j)$  estiver presente, o par  $(j, i)$  não estará presente.

O seu algoritmo deve começar a reorientação das ruas sempre a partir do vértice 0.

## Saída

Para cada caso, imprima em separado (linha-por-linha) cada rua como o par  $(i, j)$  para indicar que foi atribuída à rua a direção que vai da esquina  $i$  à esquina  $j$ . Para uma rua que não pode ser convertida em rua de mão única, imprima ambos  $(i, j)$  e  $(j, i)$  em duas linhas diferentes. A lista de ruas  $(i, j)$  deve ser impressa em ordem crescente de  $i$ . Termine cada caso com uma linha contendo um único caractere '#'.

**Observação 2:** Nesta questão, represente grafos como lista de adjacências.

## Exemplo de entrada

```
9 10
0 1
1 2
2 3
2 4
4 5
0 5
4 6
6 7
6 8
7 8
```

## Exemplo de saída

```
(0,1)
(1,2)
(2,3)
(2,4)
(3,2)
(4,5)
(4,6)
(5,0)
(6,4)
(6,7)
#
```



(7,8)

(8,6)

#

Um exemplo é ilustrado nas duas figuras abaixo.

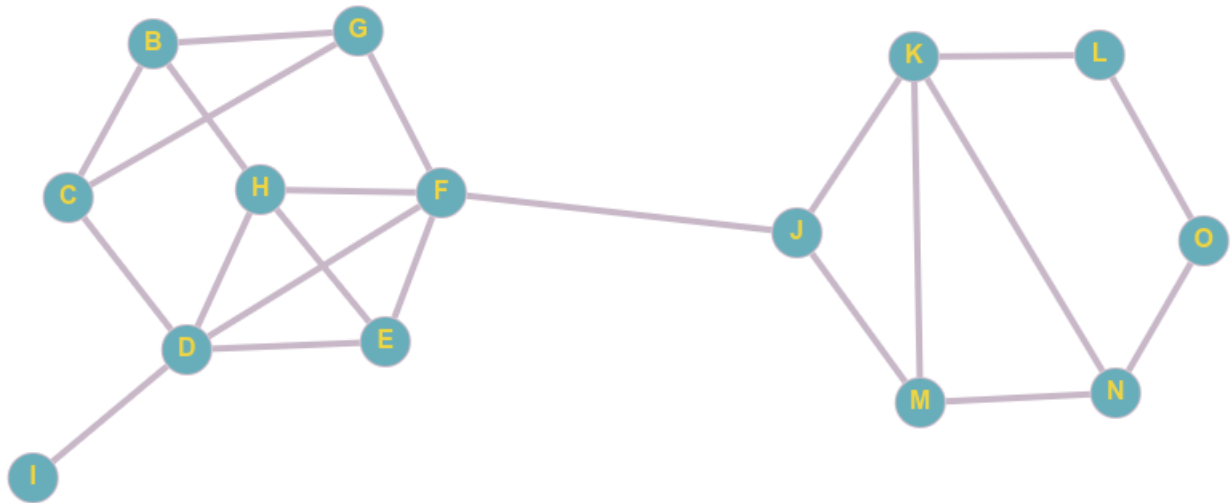


Figura 4: Grafo de uma cidade onde as ruas são todas de mão dupla. Note que a rua  $(D, I)$  é uma rua sem saída.

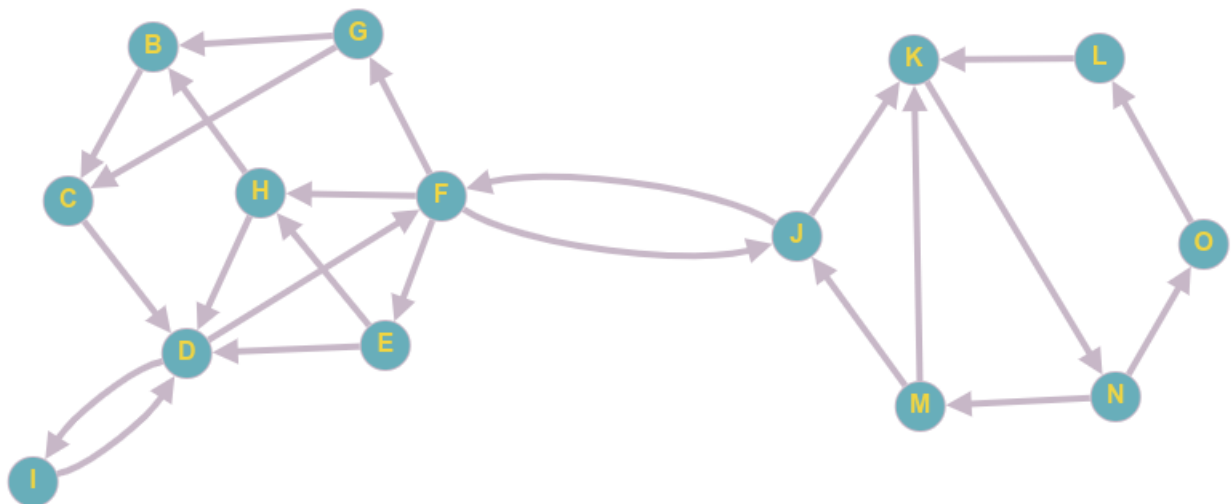


Figura 5: Grafo da figura anterior com ruas orientadas de modo que exista um caminho orientado entre quaisquer duas cidades. Note que as ruas  $(D, I)$  e  $(F, J)$  foram as únicas que permaneceram como ruas de mão dupla. Neste grafo, essas são as únicas ruas que não podem ter uma orientação de mão única.

# 1 O que deve ser entregue?

- **Código fonte em C++** (corretamente indentado e comentado). Códigos não documentados terão redução na nota.
- **Relatório do trabalho.** Para cada problema, o relatório deve conter uma descrição da solução (uma explicação de porquê a solução é correta) e da complexidade da solução em termos de tempo de execução e espaço de memória utilizado (Notação  $O$ ). Coloque a bibliografia utilizada para o desenvolvimento do trabalho. O relatório deve ser enviado em formato PDF.

## Entrega

A entrega deve ser feita via Moodle. Os arquivos de código fonte e o relatório deverão ser compactados em um único arquivo (ZIP, RAR, 7Z, etc.) e enviados.

## Informações adicionais

- Clareza, indentação e comentários no programa serão considerados na nota;
- **Não copie trabalho.** Trabalhos copiados terão nota **ZERO**.