

# MC558 - Complexidade de Algoritmos II

Primeiro semestre de 2025

## Laboratório 2

### Trilhas coloridas

Euléria, a famosa cidade natal do Prof. Sabin Ada, tem dois tipos de ruas: ruas azuis e ruas vermelhas (não me pergunte por quê...). Quando ainda era um pequeno mancebo, o Prof. Ada gostava de passear pela cidade usando as seguintes regras:

- a primeira rua pode ser azul ou vermelha,
- se ele acabou de percorrer uma rua azul, a próxima deve ser vermelha e vice-versa,
- todas as ruas devem ser percorridas exatamente uma vez, e
- ele deve voltar ao ponto de partida ao final do passeio.

A cidade e suas ruas podem ser modeladas por um grafo em que os vértices representam as interseções (cruzamentos, esquinas) das ruas, e cada aresta representa uma rua. Uma rua tem uma das cores: azul ou vermelha. Em linguagem de Teoria dos Grafos, o que o Prof. Ada queria encontrar era uma trilha euleriana em que as arestas fossem alternadamente azuis e vermelhas; chamaremos tal trilha simplesmente de **Trilha Euleriana Alternante**.

Nos primeiros anos em que começou a fazer esses passeios, ele rapidamente percebeu que o grafo correspondente a Euléria admitia uma trilha euleriana alternante. A Figura 1(a) mostra o grafo naquela época e uma possível trilha euleriana alternante. Passado algum tempo, novas ruas foram construídas na cidade, e o Prof. Ada (para sua profunda tristeza) percebeu que o novo grafo não admitia tal trilha; veja a Figura 1(b).

Sua missão é projetar um algoritmo para o problema de, dado um grafo com arestas azuis ou vermelhas, encontrar uma trilha euleriana alternante, se existir. **Sugestão:** que tal ler os slides sobre grafos eulerianos?

**Observação.** Isto não será verificado, mas sua implementação deveria ter complexidade linear.

## 1 Entrada e Saída

**Entrada:** a primeira linha contém dois inteiros  $n$  e  $m$  (os números de vértices e arestas do grafo) separados por um espaço. Cada uma das  $m$  linhas seguintes contém três números  $u$ ,  $v$  e  $c$  separados por um espaço com  $0 \leq u, v \leq n - 1$  e  $c \in \{0, 1\}$  correspondendo a uma aresta  $uv$  de cor  $c$  (0 representa a cor azul e 1 representa a cor vermelha). **Você pode supor que o grafo é simples e conexo.**

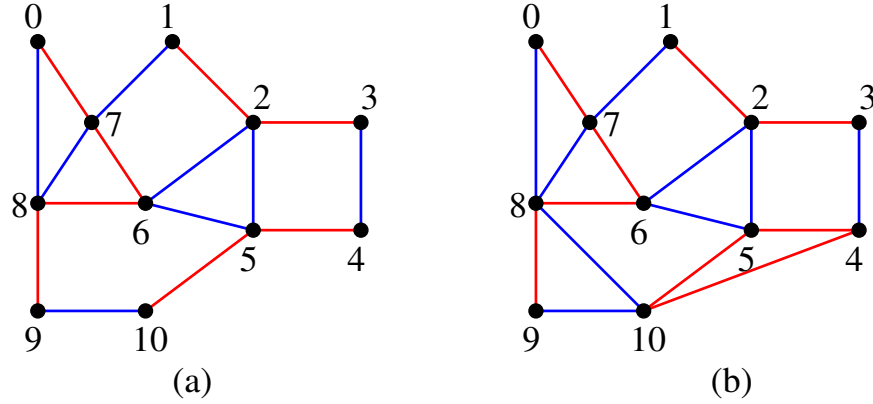


Figura 1: (a)  $T = (0, 7, 1, 2, 5, 4, 3, 2, 6, 7, 8, 6, 5, 10, 9, 8, 0)$  é uma trilha Euleriana alternante do grafo. (b) O grafo não admite uma trilha Euleriana alternante.

**Saída:** se o grafo admitir uma trilha Euleriana alternante, então imprima a sequência de vértices da trilha separados por espaço (com o primeiro vértice duplicado no final), caso contrário, imprima “Não possui trilha Euleriana alternante” (exemplos no final do arquivo).

## 2 Implementação e Submissão

- A solução deverá ser implementada em C, C++ ou Python 3.
- O programa deve ser submetido no SuSy, com o nome principal **t2** (por exemplo, t2.c).
- O número máximo de submissões é 20.
- A tarefa contém 10 testes abertos e 10 testes fechados. A nota será proporcional ao número de acertos nos testes fechados.
- **Casos de plágio implicam em nota ZERO na disciplina para todos os envolvidos.**
- **Não é permitido o uso de bibliotecas que não sejam padrão, bem como diretivas ou flags de otimização.**

## 3 Prazo final de submissão

Segunda-feira 14 de abril às 6h da manhã. (Note que a prova é dia 16 de abril)

**Exemplos:**

<b>Entrada</b>	<b>Saída</b>
11 16 0 7 1 0 8 0 1 2 1 1 7 0 2 3 1 2 5 0 2 6 0 3 4 0 4 5 1 5 6 0 5 10 1 6 7 1 6 8 1 7 8 0 8 9 1 9 10 0	0 7 1 2 5 4 3 2 6 7 8 6 5 10 9 8 0
4 5 0 2 0 2 3 1 1 0 1 3 1 0 3 0 0	Não possui trilha Euleriana alternante