

# MC558 - Complexidade de Algoritmos II

Primeiro semestre de 2025

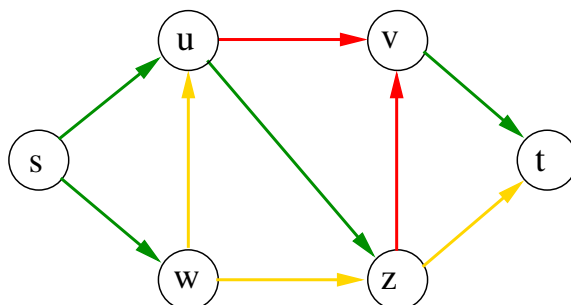
## Laboratório 3

### Muitos caminhos e um destino

Joãozinho acabou de entrar na UNICOMP. Ele está super feliz, mas agora tem que planejar sua vida nos próximos 5 anos. Na vida, há vários caminhos a escolher, que levam a vidas completamente diferentes. Desde a Antiguidade, filósofos têm meditado sobre este problema fundamental da existência, mas que agora pode ser estudado usando computadores e matemática.

O objetivo desta tarefa é calcular o número de possíveis vidas que Joãozinho pode viver, sabendo das possíveis escolhas em cada momento (obviamente, isso não é possível na vida real, mas isto é apenas um projeto de MC558). Para descrever precisamente o problema, usaremos grafos.

Seja  $G$  um grafo orientado com dois vértices especificados: um vértice origem  $s$  (início da graduação) e um vértice destino  $t$  (final da graduação). Um vértice representa um certo estado no tempo. Uma aresta  $(i, j)$  indica que é possível ir do estado  $i$  ao estado  $j$  e representa uma possível escolha no ponto/instante  $i$ . Veja a figura abaixo:



Cada aresta tem uma das 3 possíveis cores: **verde**, **amarelo** ou **vermelho**. Verde indica que foi uma escolha que representou uma boa ação (por exemplo, Joãozinho fez o projeto com antecedência!!), amarelo indica que foi um ato neutro (por exemplo, Joãozinho fez o projeto, mas em cima da hora), e vermelho significa que não foi lá essas coisas (por exemplo, Joãozinho entregou o projeto atrasado ou não entregou...). Joãozinho vem de uma família humilde, mas ele aprendeu a ser responsável desde pequeno. Assim, ele estabeleceu que as suas escolhas devem obedecer às seguintes regras:

- se ele escolheu uma aresta verde, então sua próxima aresta pode ser de qualquer cor;
- se ele escolheu uma aresta amarela, então sua próxima aresta não pode ser vermelha;
- se ele escolheu uma aresta vermelha, então sua próxima aresta tem que ser verde.

Você deve escrever um programa que recebe  $G$ ,  $s$  e  $t$  e devolve o **número de caminhos viáveis** de  $s$  a  $t$ , ou seja, caminhos de  $s$  a  $t$  que respeitem as regras acima. No grafo da figura, o número de caminhos viáveis de  $s$  a  $t$  é 6, a saber:  $(s, u, v, t)$ ,  $(s, u, z, v, t)$ ,  $(s, u, z, t)$ ,  $(s, w, u, v, t)$ ,  $(s, w, u, z, t)$  e  $(s, w, z, t)$ .

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

## 1 Entrada

Na primeira linha da entrada estão quatro inteiros  $n$ ,  $m$ ,  $s$  e  $t$ , que indicam, respectivamente, o número de vértices, o número de arestas, o vértice origem e o vértice destino de um grafo orientado acíclico  $G$ , onde  $0 \leq s, t \leq n - 1$ .

A seguir vêm  $m$  linhas, cada uma contendo uma tripla de inteiros  $x$ ,  $y$  e  $c$ , onde  $0 \leq x, y \leq n - 1$  e  $0 \leq c \leq 2$ , que indicam que  $(x, y)$  é uma aresta de  $G$  com cor  $c$  (verde = 0, amarelo = 1 e vermelho = 2). Você pode supor que:  $0 \leq n \leq 100$ , o grafo orientado é acíclico, sem arestas múltiplas, e o número de caminhos viáveis é menor que  $2^{31}$ .

## 2 Saída

Apenas um inteiro indicando o número de caminhos viáveis de  $s$  a  $t$ .

**Exemplos:**

Entrada	Saída
6 9 0 5 0 1 0 0 2 0 1 3 2 1 4 0 2 1 1 2 4 1 3 5 0 4 3 2 4 5 1	6
4 4 0 3 0 1 2 0 3 1 1 2 1 2 3 0	1

### 3 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 **t3** (por exemplo, t3.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.**

### 4 Prazo final de submissão

Segunda-feira 5 de Maio às 6h da manhã.