

Projeto e Análise de Algoritmos
Prof. Dr. Alan Demétrius Baria Valejo
Trabalho Prático 2 - Em Trenós

Lembra do jogo Among Us, mostrado nas primeiras aulas de grafos? Ele não estava lá à toda. Vamos usá-lo como contexto deste trabalho prático. Nesse jogo, todas as personagens estão em uma nave, que está passando por diversas dificuldades técnicas. Assim, o papel da tripulação é cumprir tarefas para consertar a nave e manter todos em segurança até o final da viagem. Porém, há um impostor entre os jogadores. A tarefa dele é sabotar a viagem e eliminar (meu eufemismo para “matar”) os demais jogadores. Além disso, o impostor pode entrar na tubulação de ar da nave e se locomover entre as saídas do ar rapidamente.

Neste trabalho prático você deve considerar a seguinte situação. Em um determinado momento do jogo, você vê alguém vestido de papai noel entrando na tubulação e pensa “Opa! Isso não é uma chaminé... logo, esse não é o Papai Noel de verdade. Deve ser um impostor. Ele está em trenós. Digo, entre nós!”. Sua tarefa, agora que sabe quem é o impostor, é ir até o botão de emergência para chamar uma reunião e delatar o meliante.

No entanto, o falso Noel também te viu e vai tentar te eliminar antes que você chegue ao botão. Para isso, ele pode usar os tubos de ventilação. Considere que, como ele não sabe o caminho que você vai tomar, ele vai tentar chegar na sala onde o botão se encontra e, de lá, tentar te alcançar antes que você chame a reunião.



A sua missão neste trabalho é determinar, a partir desse momento, quem consegue chegar antes na sala onde o botão de emergência se encontra. Para isso, você contará com um grafo não-direcionado e ponderado que define o mapa do jogo - onde os vértices representam as salas e as arestas, os corredores entre elas, com o peso da aresta sendo o tempo necessário para se chegar de uma sala à outra - e a definição de onde se encontram

e como se ligam as entradas da tubulação de ar. Você deve sempre considerar que o botão de emergência se encontra no vértice 0 e que se o tempo de chegada em tal vértice for igual, você tem a vantagem sobre o impostor. Ou seja, você ganha nesse caso.

Entrada:

A entrada apresenta um único grafo conexo, a descrição das tubulações e várias consultas. A primeira linha contém os valores inteiros M , E , N e C ($2 \leq M \leq 100$, $2 \leq N \leq 20$, $M \leq E \leq M^2 - M$, $2 \leq C \leq M$), representando os números de salas, de ligações entre elas por meio de corredores, de ligações entre salas por tubulações e de consultas, respectivamente. Na próxima linha, encontram-se $3 \cdot E$ valores apresentados em triplas (U, V, D) , ($0 \leq U, V < M$, $0.5 \leq D \leq 10.0$), que descrevem as ligações entre as salas U e V por meio de um corredor com comprimento D . Os indicadores de vértices são números inteiros, enquanto o valor de distância pertence aos reais e é apresentado com uma casa decimal. A seguir, são apresentados $2 \cdot N$ números inteiros, representando os pares (U, V) , ($0 \leq U, V < M$), que descrevem o fato de haver uma ligação entre as salas U e V por meio de tubos de ventilação, com distância 1 entre elas. A partir da quarta linha, há C linhas de consultas. Cada uma dessas linhas é composta por um inteiro, representando a sala onde você viu o impostor.

Saída:

Para cada uma das linhas de consulta, o seu programa deve imprimir “**defeat**” caso o impostor consiga chegar antes de sua personagem ao vértice 0 ou “**victory**”, caso contrário.

Exemplo de entrada:

```
9 12 4 2
0 1 1.3 0 5 1.5 0 4 3.2 1 2 0.7 1 3 0.3 2 3 2.3 4 9 0.2 5 6 1.7
5 7 3.0 6 7 0.9 6 8 2.1 7 8 1.1
8 7 7 5 5 8 2 3
2
8
```

Saída esperada para o exemplo:

```
victory
defeat
```

O grafo do exemplo é ilustrado a seguir:

