

Tecnologia em Jogos Digitais

Estrutura de Dados

Bombeiros

O departamento central de bombeiros de uma cidade coopera com o departamento do transporte para manter mapas da cidade que reflitam as condições atuais das ruas da cidade. Todos os dias diversas ruas são interditadas para reparos ou construção. Os bombeiros necessitam escolher rotas dos seus quartéis aos incêndios que não utilizem as ruas interditadas.

A cidade é dividida em distritos não sobrepostos, e cada um que contém um único quartel de bombeiros. Quando um incêndio é relatado, um aviso do departamento central é enviado para o quartel de bombeiro no distrito onde se encontra o incêndio. A central lhe envia também uma lista de rotas possíveis do quartel até o incêndio. Você deve escrever um programa que o departamento central dos bombeiros possa gerar as rotas possíveis do quartel ao incêndio.

Implementação

A cidade tem um mapa separado para cada distrito. A esquina de uma rua no mapa é identificada **pelos inteiros positivos menores ou iguais a 20**, com o quartel de bombeiro sempre na esquina #1.

A entrada para o programa é feita através de um arquivo, a primeira linha de consiste em um único inteiro que representa o número da esquina onde o ocorre o incêndio. As linhas seguintes consistem em uma sequência de pares de números inteiros positivos separados por espaços em branco ($x\ y$) que são as esquinas adjacentes das ruas livres (que não estão interditadas). Por exemplo, se o par **4 7** estiver em uma linha do arquivo, então a rua no sentido da esquina **4** para esquina **7** está aberta. Considere que as ruas no mapa são de mão-única. A última linha do arquivo de entrada um par de 0's informando que sequência de esquinas terminou.

Além disso, seu programa deve validar o arquivo de entrada, apresentando mensagens de erros. Como por exemplo, se existir um par $x\ y$ na sequência de pares que já foi cadastrado, então emitir uma mensagem na saída informando que o par $x\ y$ já foi inserido. Outro exemplo, se existir um par $x\ y$ na sequência e $x = y$, então emitir uma mensagem na saída informando que o x deve ser diferente do y .

A saída do seu programa deve listar as várias rotas distintas, por questões óbvias, os bombeiros não querem que seus caminhões que dirijam em círculos, sendo assim, você deve verificar, antes de cadastrar o par, se eles não formam ciclo caso formem descartar o par e mostre uma mensagem informativa. Ao final seu programa deve informar o total de rotas do quartel dos bombeiros até a esquina onde acontece o incêndio.

Exemplo de entrada

6
1 2
1 3
2 2
3 4
3 5
4 6
5 6
5 1
2 3
2 4
1 3
0 0

Exemplo de saída

Par 2 2 x deve ser diferente do y .

Par 5 1 forma ciclo

Par 1 3 já inserido

Rotas:

1 2 3 4 6

1 2 3 5 6

1 2 4 6

1 3 4 6

1 3 5 6

Existem 5 rotas do quartel de bombeiros ate a esquina 6.

Observações importantes:

O programa deve estar bem documentado e implementado na **linguagem Java** e pode ser feito em grupo de até **2 alunos**, não esqueçam de colocar o **nome dos integrantes** do grupo no programa fonte. A entrega do trabalho deve ser feita pelo *Blackboard* (**não serão aceitos trabalhos entregues via e-mail**) e será avaliado de acordo com os seguintes critérios:

- O problema deve utilizar a classe **Grafo** (conforme visto em sala) que armazena o grafo em matrizes de adjacências.
- Para verificação do ciclo e o cálculo de todos as rotas possíveis utilize **o algoritmo de busca em profundidade** visto em sala.
- Utilização dos conhecimentos de orientação a objetos vistos na disciplina de Projeto Integrador III.
- O quão fiel é o programa quanto à descrição do enunciado;
- Funcionamento do programa, Indentação, comentários e legibilidade do código e clareza na nomenclatura de variáveis;

Como este trabalho pode ser feito em **grupo**, evidentemente você pode “*discutir*” o problema dado com outros **grupos**, inclusive as “*dicas*” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa. Ou seja, qualquer tentativa de fraude será punida com a **nota zero**.