TRABALHO PRÁTICO
TEORIA DE GRAFOS 2018/1

Professor: Rafael Santos Coelho
26 de maio de 2018

Descrição do problema
Uma importante metrópole brasileira vem recentemente recebendo ameaças de um grupo terrorista. Membros desse grupo são conhecidos internacionalmente por plantarem bombas em lugares de alta circulação de pessoas. Em geral, os ataques sempre acontecem da seguinte maneira: instantes antes de acionarem o explosivo, os terroristas, em um ato de sadismo, informam autoridades do governo a localização e o "raio" de destruição da bomba. Dessa maneira, cabe à prefeitura da cidade, conhecendo o mapa da cidade, executar um plano emergencial de evacuação para que ninguém se machuque. Você foi contratado para projetar um algoritmo que, dados o mapa da cidade, a localização da bomba e seu "raio" de destruição, calcula exatamente os pontos da cidade que serão atingidos pela bomba e que, portanto, precisam ser evacuados.

Entrada
Os dados de entrada **deverão ser lidos do teclado exatamente no formato descrito a seguir**. O *input* começa com uma linha contendo um número **N (um inteiro positivo)** de casos de teste. Nas linhas seguintes aparecem os N casos de teste. Cada caso de teste é descrito por uma sequência de informações especificadas em uma determinada ordem. Primeiramente, é informado o mapa da cidade. Para isso, são passados em uma linha dois números, um número **L (um inteiro positivo)** que denota o número de **localidades da cidade** e um número **R (um inteiro positivo)** que denota o número de **ruas da cidade**. Na linha seguinte, são passadas informações a respeito das ruas da cidade. Para cada rua, é passada uma tripla de números **i j d** (**cada número em uma mesma tripla separado do outro por um espaço em branco e cada tripla separada da outra por um espaço em branco também**), onde **i** e **j (números inteiros positivos)** são números de localidades no mapa da cidade e **d (um inteiro positivo)** é o comprimento em metros da rua (ou avenida) que liga as localidades **i** e **j**. Na linha seguinte, é passado um par de números **k r**, onde **k (um inteiro positivo)** é o **número da localidade no mapa da cidade onde a bomba foi plantada** e **r (um inteiro positivo) é o "raio" de destruição em metros da bomba**. **Considere que uma localidade da cidade precisa ser evacuada se estiver a uma distância de no máximo r metros da localidade de número k (obviamente, tal distância leva em consideração as distâncias das ruas que interligam diferentes localidades no mapa da cidade)**.

Saída
Para cada caso de teste, seu algoritmo **deve imprimir na tela** (**exatamente da forma descrita a seguir**) uma mensagem (**sem as aspas, é claro**) iniciando com "*Caso p: "*, onde p é o número do caso de teste em questão (**começando a contagem a partir de 1 e seguindo a ordem natural sucessivamente**). Após isso, você deve informar, **em ordem crescente**, os números das localidades no mapa da cidade **que precisam ser evacuados** (**separe um número do outro por um espaço em branco**).

Exemplo de entrada (comentários escritos abaixo em rosa explicam cada linha)
2 **<= Número de casos de teste**
10 10 **<= Número de localidades e de ruas do caso de teste 1**
1 9 7 1 2 8 1 8 5 1 5 2 5 6 30 3 5 4 6 8 30 8 10 13 7 10 21 4 7 2 **<= Ruas no caso de teste 1**
5 30 **<= Localização da bomba e "raio" de destruição no caso de teste 1**
7 9 **<= Número de localidades e de ruas do caso de teste 2**
1 7 3 1 6 55 1 2 10 1 5 44 5 6 9 3 5 27 2 7 30 2 3 30 4 5 13 **<= Ruas no caso de teste 2**
2 40 **<= Localização da bomba e "raio" de destruição no caso de teste 2**

Exemplo de saída (relativo ao exemplo de entrada exibido acima)
Caso 1: 1 2 3 5 6 8 9 10
Caso 2: 1 2 3 7

Sobre a entrega e sobre os critérios de correção do trabalho

- Este trabalho poderá ser feito em grupos **de no máximo 6 integrantes**.
- Trabalhos feitos por grupos com mais de 6 integrantes **não serão corrigidos** e, portanto, **ficarão com nota 0**.
- **As únicas linguagens de programação permitidas para resolução deste trabalho são Python, C, C++ e Java**. Trabalhos feitos em qualquer outra linguagem que não seja alguma dessas **não serão corrigidos** e, portanto, **ficarão com nota 0**.
- Os códigos deverão ser entregues via Moodle até as 12h (meio dia) do dia 20/06/18 (20 de junho de 2018, uma quarta-feira). **Apenas um membro de cada grupo deve submeter o trabalho em nome do grupo inteiro**. O código-fonte deve ser compactado em um arquivo **.zip** e apenas o arquivo compactado deverá ser enviado via Moodle. **Submissões atrasadas (não importa quão pequeno seja o atraso ou o motivo do atraso) não serão aceitas.**
- **Os trabalhos serão corrigidos "ao vivo" junto com os alunos durante a aula do dia 20/06/18**. Cada grupo será chamado e arguido pelo professor. O professor vai fazer testes para verificar a corretude do código e vai fazer perguntas para cada membro do grupo a respeito do código feito. Portanto, é essencial que cada integrante do grupo entenda **profundamente** o código e saiba explicá-lo detalhadamente.
- A nota dada será individual para cada membro do grupo e vai ter dois componentes: um objetivo (valendo 70% do total, que vai depender das respostas que o código deu para os testes feitos) e um subjetivo (valendo 30% do total, que vai depender de quão bem cada integrante saiu ao ser questionado a respeito de determinadas partes do código).
- **Pessoas que faltarem a aula do dia 20/06/18 e não justificarem devidamente a falta ficarão com nota 0.**
- **Códigos mal documentados, mal indentados e obscuros poderão ser penalizados também.**
- **Para cada caso teste, será fixado um limite máximo de tempo de execução (esse limite será razoavelmente folgado). Códigos corretos que estourarem esse limite serão penalizados.**
- **Qualquer tentativa de fraude (plágios) identificada poderá acarretar nota 0 aos envolvidos (isto é, plagiados e plagiadores).**
- **Códigos com erros de compilação ficarão com nota 0. Códigos com erros de execução (por exemplo, falha de segmentação) serão penalizados.**
- **Para resolver o trabalho, é necessário usar pelo menos um dos algoritmos de grafos ensinados em sala**