

INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos

2013/2014

2^o Projecto

Data Limite de Entrega: 23 de Abril de 2014

O Problema

A segurança é uma preocupação importante em vários tipos de eventos, particularmente aqueles que envolvem grandes grupos de pessoas, multidões. Eventos como jogos de futebol ou manifestações exigem bastante trabalho logístico por parte das forças de segurança envolvidas.

Neste trabalho iremos considerar um modelo simples de controlo de multidões. Consideramos o mapa de uma cidade, onde existem pontos, onde as pessoas se podem concentrar e ligações entre esses pontos. Dois pontos podem estar ligados, no máximo, de uma forma. A nossa preocupação é garantir que não há comunicação, entre um conjunto de pontos críticos. Se houver comunicação entre todos os pontos críticos é mais provável que seja iniciado um motim, num desses pontos. Se tal acontecer a desordem irá estender-se a todos os outros pontos que lhes estejam directamente ligados, e assim sucessivamente a todos os pontos que sejam atingíveis a partir do ponto inicial.

Para evitar comprometer a segurança da cidade inteira as forças de segurança podem barrar algumas das ligações, por forma a evitar que seja iniciado um motim. Para minimizar os custos desta operação, queremos minimizar o número de ligações que é necessário barrar. Assim o objectivo deste projecto consiste em modelar este problema, utilizando estruturas de dados e algoritmos adequados, por forma a que, dado o mapa de uma cidade e uma lista de pontos críticos, o algoritmo calcule o número mínimo de ligações que é necessário barrar. O objectivo é que os pontos críticos não estejam todos em comunicação. Queremos que pelo menos um conjunto não vazio de pontos críticos fique isolado dos restantes, minimizando o número de ligações a barrar.

Input

A primeira linha do input contém dois inteiros. O primeiro indica o número de pontos n e o segundo inteiro o número de ligações m .

Seguem-se as ligações, uma por linha. Cada linha contém dois inteiros, que identificam os pontos que estão ligados. Estes números u e v respeitam as seguintes restrições: $0 \leq u < v < n$.

Após as linhas das ligações existe uma linha que contém apenas um número h , a indicar o número de problemas que se seguem. Todos estes problemas, i.e., conjuntos de pontos, devem ser resolvidos sobre o grafo lido anteriormente. As seguintes h linhas contém a listas de pontos críticos. O primeiro número de cada linha, k indica quantos pontos críticos existem nessa linha, com $1 \leq k \leq n$. Os seguintes k números são os dos pontos críticos.

Todos os números que esteja numa mesma linha são separados por um espaço em branco.

Output

Para cada caso de testes o output deve consistir em h linhas, cada uma com apenas um número, o número mínimo de ligações a barrar, para o respectivo problema. A ordem dos números deve ser a dada no input. Cada linha de output não deve conter mais nenhum caracter, para além dos dígitos da resposta e do ' $\backslash n$ '.

Exemplos

input 1

```
4 4
0 2
0 3
1 2
1 3
2
2 0 1
2 0 2
```

output 1

```
2
2
```

input 2

```
5 5
0 2
0 3
1 2
1 3
0 4
1
3 0 1 4
```

output 2

```
1
```

input 3

```
6 4
0 4
0 5
1 4
1 5
1
2 0 1
```

output 3

```
2
```

Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões em linguagem Java não serão aceites.

O tempo necessário para implementar este projecto é inferior a 20 horas.

Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Informação sobre as linguagens de programação possíveis está disponível no website do sistema Mooshak. A linguagem de programação é identificada pela extensão do ficheiro.

Por exemplo, um projecto escrito em `c` deverá ter a extensão `.c`. Após a compilação, o programa resultante deverá ler do 'standard input' e escrever para o 'standard output'. Informação sobre as opções e restrições de compilação podem ser obtidas através do botão 'help' do sistema Mooshak. O comando de compilação não deverá produzir output, caso contrário será considerado um erro de compilação. O relatório deverá ser entregue no formato PDF com não mais de 4 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma introdução breve, a descrição da solução, a análise teórica e a avaliação experimental dos resultados. O relatório deverá incluir qualquer referência que tenha sido utilizada na realização do projecto. Relatórios que não sejam entregues em formato PDF terão nota 0. O código fonte deve ser submetido através do sistema Mooshak e o relatório (em formato PDF) deverá ser submetido através do Fénix. O código fonte será avaliado automaticamente pelo sistema Mooshak. Observe que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas; tal inclui o código fonte e o relatório.

Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note que também é possível submeter várias vezes no Fénix e que não serão aceites relatórios fora de prazo e não haverá extensão de prazo.

O sistema Mooshak indica o tempo disponível para o projecto ser submetido. Os projectos têm que ser submetidos para o sistema Mooshak; não existe outra forma de submissão do projecto. Os relatórios têm que ser submetidos no sistema Fénix; não existe outra forma de submissão dos relatórios.

Avaliação

O projecto deverá ser realizado em grupos de um ou dois alunos e será avaliado em duas fases. Na primeira fase, durante a submissão, cada implementação será executada num conjunto de testes, os quais representam 80% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 20% da nota final.

Avaliação Automática

A primeira fase do projecto é avaliada automaticamente com um conjunto de testes, os quais são executados num computador com o sistema operativo **GNU/Linux**. É essencial que o código fonte compile sem erros e respeite os standards de entrada e saída indicados anteriormente. Os projectos que não respeitem os formatos especificados serão penalizados e poderão ter nota 0, caso falhem todos os testes. Um conjunto reduzido de testes utilizados pelo sistema Mooshak serão públicos. A maior parte dos testes **não** serão divulgados antes da submissão. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projecto. Além de verificar a correcção do output produzido, o ambiente de avaliação restringe a memória e o tempo de execução disponíveis. A maior parte dos testes executa o comando `diff` da forma seguinte:

```
diff output result
```

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `output` contém o valor esperado. Um programa passa num teste quando, e recebe o valor correspondente, quando o comando `diff` não reporta quaisquer diferenças. Existem 16 testes. Assim, o sistema reporta um valor entre 0 e 16.

Deteção de Cópias

A avaliação dos projectos inclui um procedimento para deteção de cópias. A submissão de um projecto implica um compromisso de que o trabalho foi realizado exclusivamente pelo aluno. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo aluno implica a reprovação na unidade curricular, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em “Diário da República”.