

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

## 2018/2019

### 1<sup>o</sup> Projecto

Data Limite de Entrega: 22 de Março de 2019

## Descrição do Problema

O Sr. João Caracol está a gerir uma rede que foi crescendo de forma desordenada. Actualmente, é necessário fazer uma auditoria à rede e determinar várias propriedades. Em princípio, todos os routers da rede estão ligados entre si, ou seja, consegue-se definir um caminho que permite a comunicação entre qualquer par de routers da rede. Caso isso não aconteça, então a rede está dividida em sub-redes (sub-conjuntos de routers ligados entre si).

Para além da identificação de sub-redes, o Sr. João quer identificar se a sua rede é robusta. Em particular, ele pretende identificar os routers da rede que se forem atacados (não permitindo a recepção ou envio de informação) ou desligados, resultariam no aumento do número de sub-redes.

O seu objectivo é fazer um programa que ajuda o Sr. João Caracol a fazer a auditoria à rede.

## Input

O ficheiro de entrada descreve a rede do Sr. João Caracol. Cada router é identificado por um identificador inteiro entre 1 e  $N$ , sendo  $N$  o número de routers na rede. As ligações entre dois routers são bi-direccionais, ou seja, se há uma ligação entre os routers  $r_1$  e  $r_2$ , então os dados podem ser transmitidos em ambos os sentidos.

O input é definido da seguinte forma:

- Uma linha com o número de routers da rede  $N$  ( $N \geq 2$ ).
- Uma linha com o número de ligações entre routers na rede  $M$  ( $M \geq 1$ ).

- Uma lista de  $M$  linhas, em que cada linha contém dois inteiros  $u$  e  $v$  (separados por um espaço em branco) indicando que há ligação entre  $u$  e  $v$ .

## Output

No output, uma sub-rede é identificada pelo router com **maior** identificador que pertence à sub-rede. O seu programa deverá escrever no output a seguinte informação:

- Uma linha com um inteiro  $R$  que denota o número de sub-redes.
- Uma linha com  $R$  valores inteiros separados por espaços em branco com os identificadores das  $R$  sub-redes. Nesta linha, os  $R$  valores têm que aparecer ordenados de forma crescente.
- Uma linha com um inteiro  $C$  que denota o número de routers que quebram uma sub-rede. Ou seja, o número de routers que se removidos individualmente, resultaria no aumento de sub-redes.
- Uma linha com um inteiro  $m$  que denota o número de routers da maior sub-rede resultante da remoção de todos os  $C$  routers que quebram uma sub-rede.

## Exemplos

### input 1

```
3
2
1 2
2 3
```

### output 1

```
1
3
1
1
```

### input 2

```
6
6
1 2
2 3
```

```
3 1
1 4
5 4
6 4
```

### **output 2**

```
1
6
2
2
```

### **input 3**

```
8
7
1 2
2 3
3 1
2 4
5 6
5 7
6 8
```

### **output 3**

```
2
4 8
3
2
```

## **Implementação**

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões em linguagem Java também são aceitáveis, devendo no entanto ter particular atenção a aspectos de implementação.

O tempo estimado para implementar este projecto é inferior a 15 horas.

## Submissão do Projecto

A submissão do projecto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Serão utilizadas as plataformas Mooshak para a submissão do código e Fénix para submissão do relatório. 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. O texto do relatório deve ser cuidado. Serão aplicados descontos por erros gramaticais ou erros ortográficos. 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 output esperado. Um programa passa num teste e recebe o valor correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output).

A nota obtida na classificação automática poderá sofrer eventuais cortes caso a análise do código demonstre recurso a soluções ajustadas a inputs concretos ou outputs aleatórios/constantes.

## Detecção de Cópias

A avaliação dos projectos inclui um procedimento para detecção de cópias através do sistema moss (<https://theory.stanford.edu/~aiken/moss/>). A submissão de um projecto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a reprovação na unidade curricular, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa ou 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”.