

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

## 2014/2015

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

Data Limite de Entrega: 23 de Março de 2014

## O Problema

Paul Erdős (1913-1996) foi reconhecidamente um dos maiores matemáticos do século XX. Para além das suas valiosas contribuições em teoria dos conjuntos, teoria dos números ou teoria dos grafos (entre outras áreas), Erdős ficou conhecido por ser autor de mais de 1500 artigos científicos. Para tal, Erdős colaborou com mais de 500 outros matemáticos na co-autoria desses artigos.

Dada a extensão das colaborações de Paul Erdős, tornou-se popular calcular o número de Erdős dos cientistas. O número de Erdős é definido como sendo a distância colaborativa de uma pessoa a Paul Erdős. Considera-se que os co-autores de artigos com Paul Erdős estão à distância de 1. Os co-autores de artigos com co-autores de Paul Erdős estão à distância de 2, e assim sucessivamente. Considera-se que o número de Erdős de Paul Erdős é 0.

## Input

O ficheiro de entrada deverá conter a informação sobre colaborações entre cientistas. Considera-se que um cientista colabora com outro se são co-autores de pelo menos um artigo científico. O input é definido da seguinte forma:

- Uma linha com o número de pessoas  $N$  ( $N \geq 2$ ) e o número de relações de colaboração  $C$  separados por um espaço em branco.
- Uma linha com o número entre 1 e  $N$  que identifica Paul Erdős.

- Uma lista de  $C$  linhas, em que cada linha contém dois inteiros  $u$  e  $v$  (separados por um espaço em branco) indicando que a pessoa identificada por  $u$  tem pelo menos uma co-autoria com a pessoa identificada por  $v$ .

Assume-se que a identificação das pessoas é um inteiro entre 1 e  $N$  e que todas as  $N$  pessoas estão a uma distância colaborativa finita de Paul Erdős.

## Output

Suponha que  $M$  é o maior valor de número de Erdős que se deduz do input. O programa deverá escrever no output a seguinte informação:

- Uma linha com o valor  $M$ .
- Uma sequência de  $M$  linhas em que a linha  $i$  ( $1 \leq i \leq M$ ) corresponde ao número de pessoas com número de Erdős igual a  $i$ .

## Exemplos

### input 1

```
8 9
1
1 2
1 3
3 4
4 2
2 5
5 6
6 7
7 5
4 8
```

### output 1

```
3
2
2
3
```

## **input 2**

```
8 9
3
1 2
1 3
3 4
4 2
2 5
5 6
6 7
7 5
6 8
```

## **output 2**

```
5
2
1
1
2
1
```

## **Implementação**

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões em linguagem Java também são aceitáveis, mas devem ter mais cuidado com alguns aspectos de implementação.

O tempo necessário para implementar este projecto é inferior a 10 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 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). Existem 16 testes. Assim, o sistema reporta um valor entre 0 e 16.

## **Detecção de Cópias**

A avaliação dos projectos inclui um procedimento para detecção de cópias. 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 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”.