

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

2021/2022

## 2º Projecto

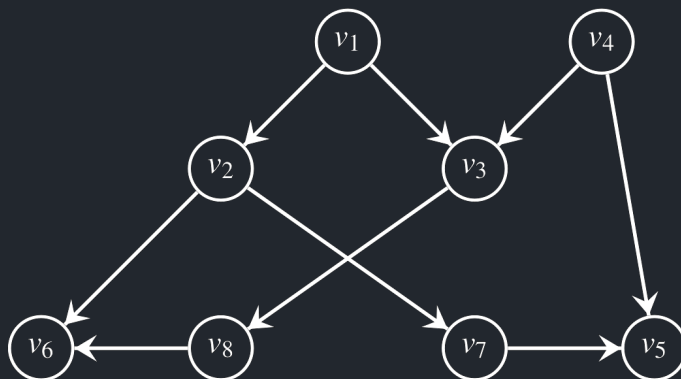
Data enunciado: 14 de Janeiro de 2022  
Data Limite de Entrega: 28 de Janeiro de 2022 (17h00)

### Descrição do Problema

Uma árvore genealógica consiste num grafo dirigido em que cada nó representa uma pessoa e os vizinhos directos de um nó correspondem aos seus filhos, podendo existir nós orfãos, nós com um progenitor e nós com dois progenitores. Dados dois nós  $P1$  e  $P2$  de uma árvore genealógica:

- $P1$  diz-se *ancestral* de  $P2$  se é possível atingir  $P2$  a partir de  $P1$  na árvore genealógica;
- $P3$  diz-se um *ancestral comum mais próximo* de  $P1$  e  $P2$ , se é ancestral de  $P1$  e  $P2$ , e não existe um nó  $P4$  descendente de  $P3$  que seja também ancestral de  $P1$  e  $P2$ .

Observamos que a definição de ancestral comum mais próximo permite que dois nós tenham vários ancestrais comuns mais próximos. Por exemplo, no grafo abaixo,  $v_2$  e  $v_4$  são ambos ancestrais comuns mais próximos de  $v_5$  e  $v_6$ .



Dado um grafo dirigido  $G = (V, E)$  e dois vértices  $v_1, v_2 \in V$ , pretende determinar-se: **(1)** se  $G$  forma uma árvore genealógica válida e, caso forme, **(2)** o conjunto de ancestrais comuns mais próximos entre  $v_1$  e  $v_2$ .

## Input

O ficheiro de entrada contém a informação relativa à árvore genealógica a ser processada e aos vértices  $v_1$  e  $v_2$ , cujos ancestrais comuns mais próximos devem ser calculados, e é definido da seguinte forma:

- uma linha contendo os identificadores dos vértices  $v_1$  e  $v_2$ ;
- uma linha contendo dois inteiros: o número  $n$  de vértices ( $n \geq 1$ ), e o número  $m$  de arcos ( $m \geq 0$ );
- uma lista de  $m$  linhas, em que cada linha contém os identificadores dos vértices  $x$  e  $y$ , indicando que  $y$  é filho de  $x$ .

Os identificadores dos vértices são números inteiros entre 1 e  $n$ .

## Output

O programa deverá escrever no output “0” se o grafo não formar uma árvore válida. Caso contrário, deverá escrever no output a sequência dos identificadores de todos os ancestrais comuns mais próximos, ordenados por valor crescente e separados por um espaço em branco (termine com um espaço em branco no fim para facilitar). Caso não exista nenhum, deverá escrever no output “-”.

## Exemplo

### Input 1

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

7 5  
8 6

### Output 1

2 4

### Input 2

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

### Output 2

2

### Input 3

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

### Output 3

—

## Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões nas linguagens Java/Python também serão aceites, embora fortemente desaconselhadas. Alunos que o escolham fazer devem estar cientes de que submissões em Java/Python podem não passar todos os testes mesmo implementando o algoritmo correcto.

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

### Parâmetros de compilação:

```
C++: g++ -std=c++11 -O3 -Wall file.cpp -lm
C: gcc -O3 -ansi -Wall file.c -lm
Javac: javac File.java
Java: java -Xss32m -Xmx256m -classpath . File
Python: python3 file.py
```

## 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.

**Relatório:** deverá ser submetido através do sistema Fénix no formato PDF com não mais de 2 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma 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. Atempadamente será divulgado um template do relatório.

**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 (<http://acp.tecnico.ulisboa.pt/~mooshak/>). Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note 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.

## 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 85% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 15% 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. Os 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). O sistema reporta um valor entre 0 e 170.

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.

### Deteção 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”.