

# Relatório 2º projecto ASA 2021/2022

**Grupo:** al021

**Aluno(s):** Ana Jin (99176) e Inês Pissarra (99236)

---

## Descrição do Problema e da Solução

**Problema:** Dado um grafo dirigido, que representa uma árvore genealógica, quer-se encontrar os ancestrais comuns mais próximos entre dois dos seus nós. Isto significa que o ancestral consegue atingir ambos os nós e não tem nenhum outro descendente que também o faça. Para o grafo ser considerado uma árvore genealógica não pode conter ciclos e nem pode existir um nó com mais de dois progenitores.

**Solução:** Primeiro é necessário verificar se cada nó tem, no máximo, dois progenitores (e para tal basta contar quantos arcos apontam para cada nó) e se o grafo dado não tem ciclos. Para este último problema é possível correr uma DFS (Depth-First Search), e o grafo não é acíclico caso se pretenda visitar um nó que ainda não foi “fechado”. Para a segunda parte do problema (encontrar ancestrais comuns) é mais simples que os arcos apontem dos filhos para os pais. Pode ser utilizada uma BFS (Breadth-First Search) a partir do primeiro nó, de modo a descobrir quais os outros nós que podem ser atingidos por este. É por fim feita uma segunda BFS a partir do segundo nó, onde são logo marcados os ancestrais comuns mais próximos e os ancestrais destes (estes últimos marcados de forma diferente dos primeiros pois já não fazem parte da solução).

## Análise Teórica

Sendo  $V$  a quantidade de nós e  $E$  a quantidade de arcos existentes no grafo:

- Leitura dos dados de entrada: simples leitura do input, com um ciclo a depender de linearmente de  $E$ . Logo,  $\Theta(E)$
- Inicialização de um vetor de tamanho  $V$  colocando valores todos a 0. Logo,  $O(V)$
- Aplicação do algoritmo para para verificação da validade da árvore genealógica: a DFS é invocada uma vez por cada nó, e contém um ciclo que irá ser executado, no total, uma vez por cada arco do grafo. Contém também a inicialização de um vetor de tamanho  $V$  colocando valores todos a 0. Logo,  $O(V+E)$
- Aplicação do algoritmo que obtém os ancestrais: contém a inicialização de três vetores de tamanho  $V$  e duas BFS's. Para cada BFS, cada nó é inserido, no máximo, uma vez na fila de prioridade; e contém um ciclo que é executado, no máximo, uma vez por cada arco do grafo. Logo,  $O(V+E)$
- Apresentação dos dados: contém um ciclo a depender linearmente de  $V$ . Logo,  $O(V)$

Complexidade global da solução:  $O(V+E)$

# Relatório 2º projecto ASA 2021/2022

**Grupo:** al021

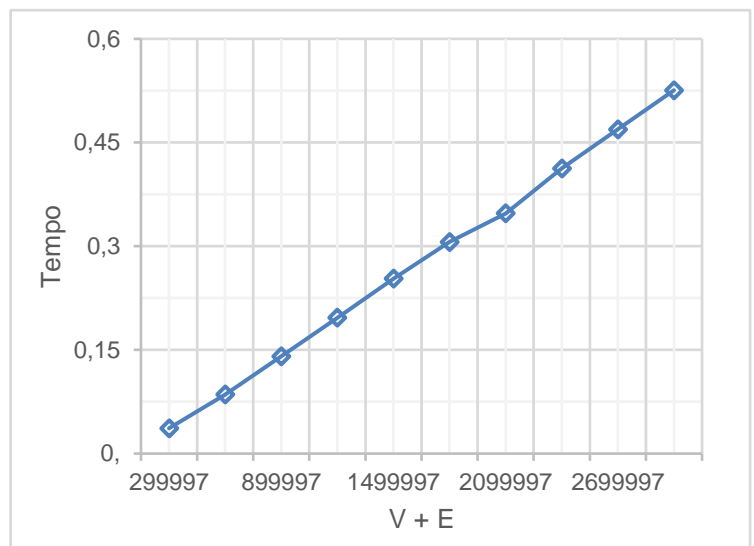
**Aluno(s):** Ana Jin (99176) e Inês Pissarra (99236)

---

## Avaliação Experimental dos Resultados

Os ficheiros testes foram gerados a partir do ficheiro randGeneoTree.cpp. Os argumentos foram os seguintes: (número de vértices) 0.9, tendo sido o número de vértices substituído pelos valores 100000, 200000, 300000, 400000, 500000, 600000, 700000, 800000, 900000 e 1000000. Nos seguintes gráficos o eixo YY representa o valor do tempo que o programa demorou executar os testes e o eixo XX representa a soma do número de nós (V) e do número de arcos (E).

V	E	V + E	Tempo (s)
100000	199997	299997	0.036625
200000	399997	599997	0.085375
300000	599997	899997	0.14075
400000	799997	1199997	0.1965
500000	999997	1499997	0.253125
600000	1199996	1799996	0.306
700000	1399997	2099997	0.3475
800000	1599997	2399997	0.412625
900000	1799997	2699997	0.469
1000000	1999996	2999996	0.525625



A partir deste gráfico, podemos concluir que a análise teórica prevista está de acordo com a avaliação experimental dos resultados.