

Relatório 2º projecto ASA 2021/2022

Grupo: tp019

Aluno(s): André Torres (99053) e Gonçalo Nunes (99074)

Descrição do Problema

O projeto consiste em verificar a validade de uma dada árvore genealógica e, posteriormente, encontrar os ancestrais comuns mais próximos, entre dois vértices, na árvore introduzida, caso estes existam.

Uma árvore genealógica diz-se válida se não existirem ciclos entre os vértices da árvore, sendo que cada vértice pode apenas ter dois ou menos vértices “pais”, isto significa que podem existir vértices isolados dos restantes. Os ancestrais comuns mais próximos caracterizam-se por serem os ancestrais comuns que não tenham descendentes que também sejam ancestrais dos vértices alvo.

Descrição da Solução

A solução baseia-se em representar a árvore genealógica através de um grafo, onde cada vértice “filho” tem conhecimento dos seus “pais”, isto é, um grafo dirigido representado através de uma lista de adjacências. Contudo, cada aresta tem o sentido oposto ao esperado numa árvore genealógica habitual, é como se estivéssemos a representar a árvore inversa à árvore original.

De forma a verificar a validade da árvore dada, temos que verificar se cada vértice tem no máximo dois vizinhos diretos, e se o grafo construído não tem ciclos. A primeira verificação é realizada na construção da lista de adjacências que representa o grafo. Para a segunda, é feita uma *Depth-First-Search* onde cada vértice apenas pode ser visitado uma vez, caso isso não aconteça, estamos perante um grafo cíclico e o programa termina.

De seguida, determinamos quais os ancestrais comuns aos vértices alvo. Para isso, executamos duas *Depth-first-search visits* a ambos os vértices. A primeira, irá marcar os ancestrais do primeiro vértice. A segunda irá marcar os ancestrais previamente marcados pela primeira, como sendo ancestrais comuns a ambos os vértices.

Resta-nos, portanto, encontrar de entre os ancestrais comuns, quais são os mais próximos. Sendo assim, construímos um sub-grafo com apenas os vértices previamente classificados como sendo ancestrais comuns. Este sub-grafo irá ter as suas arestas invertidas pois, desta forma, os vértices que não tiverem vizinhos diretos são os ancestrais comuns mais próximos que procuramos. Sendo assim, basta percorrer os vertices deste sub-grafo e imprimir no terminal os vertices que não tenham vizinhos diretos, concluindo, assim, a resolução do problema.

Relatório 2º projecto ASA 2021/2022

Grupo: tp019

Aluno(s): André Torres (99053) e Gonçalo Nunes (99074)

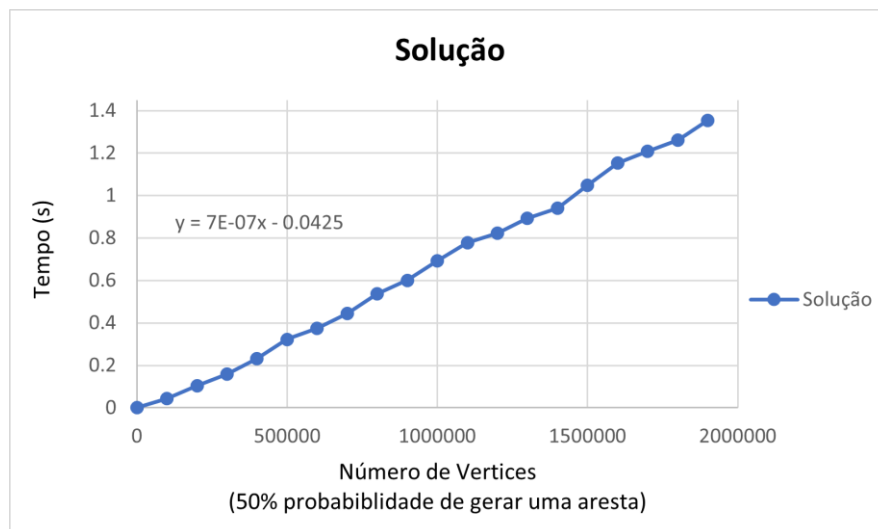
Análise Teórica

- Leitura dos dados de entrada e construção do grafo: $O(E)$
- *DFS* para verificar se o grafo é acíclico: $O(V+E)$
- Aplicação de uma *DFS-Visit* a cada um dos vértices alvo: $O(V+E)$
- Construção do sub-grafo apenas com os ancestrais comuns: $O(V+E)$
- Encontrar os vértices do subgrafo sem vizinhos diretos e apresentação dos resultados: $O(V)$

Complexidade global da solução: $O(V+E)^1$.

Avaliação Experimental dos Resultados

Para este problema geramos 20 árvores aleatórias com o número de vértices a variar entre 0 e 1900000, com incrementos de 100000 e com uma probabilidade de cada vértice gerar uma aresta de 50%. Medimos o tempo de execução do programa com cada um dos *inputs* gerados e obtivemos o seguinte gráfico.



O gráfico gerado está em concordância com a análise teórica prevista, ou seja, um crescimento de complexidade linear, que pode ser verificada pela regressão linear obtida.

Referências

- Cormen T. et al (2009). *Introduction to Algorithms*. MIT Press

¹ V corresponde ao número de vértices e E ao número de arestas (*edges*) do grafo que representa a árvore geneológica.