

Relatório 1º projeto ASA 2021/2022

Grupo: al006

Alunos : João Costa (99985) e Tomás Mendonça (96338)

Descrição do Problema e da Solução

Problema 0:

Verificar se o grafo dado é uma árvore genealógica.

Foram identificadas duas condições:

1. Cada nó tem no máximo dois pais, verificada em tempo linear, contando o número de arcos que acaba no nó i , através de uma array e verificando que esse contador nunca passa de dois.
2. Não pode haver ciclos, verificada através de uma procura DFS com stack. A DFS é aplicada começando em cada nó, mas ignoram-se nós já explorados.

Problema 1:

Encontrar o(s) antepassado(s) comum(s) mais próximo(s) (LCA).

Para tal, utiliza-se uma representação do grafo invertido e aplica-se uma pesquisa BFS a cada vértice do problema, acabando com dois vetores que representam todos os antepassados de cada vértice. Juntando os dois obtemos os antepassados comuns e, por fim, analisa-se o vetor final de forma a encontrar que nós nesse vetor é que não têm “filhos” também no vetor, pelo que seriam os antepassados comuns mais próximos possíveis.

Análise Teórica

Recolha e tratamento de dados:

A recolha é feita em tempo linear, utilizando um scanf e organizando diretamente os arcos num vetor de vetores.

É de notar que logo nesta fase se cria o grafo invertido, pois vai ser útil no futuro e evita-se um tratamento de tempo linear ligeiramente pior mais tarde.

Visto que a introdução de um arco na estrutura de dados é $O(1)$, o tratamento será feito em $O(2E)$, ou seja, em $O(E)$.

Problema 0:

A contagem e verificação para a condição 1 é feita em simultâneo com a recolha de dados e só implica duas operações em cada arco, pelo que a complexidade temporal é igual à de recolha de dados $O(E)$ e a espacial é $O(V)$.

A condição 2 implica uma DFS completa, isto é, como é iniciada para todos os nós, mas são ignorados os nós previamente explorados, a DFS é, na prática, o pior caso, porque passa obrigatoriamente por todos os nós e analisa todos os arcos. Sendo a sua complexidade temporal $O(V + E)$.

Em termos de complexidade espacial, utiliza-se uma stack de tamanho máximo $V - 1$ e um vetor *visited* de tamanho V , ou seja, complexidade $O(V)$.

Relatório 1º projeto ASA 2021/2022

Grupo: al006

Alunos : João Costa (99985) e Tomás Mendonça (96338)

Problema 1:

O problema 1 consiste na inversão do grafo, aplicação de duas BFS e verificação de vetores de tamanho V .

A inversão do grafo já foi feita na recolha de dados.

De seguida, as BFS não têm qualquer adaptação ao problema, pelo que têm complexidade temporal $O(V + E)$ e espacial $O(V)$.

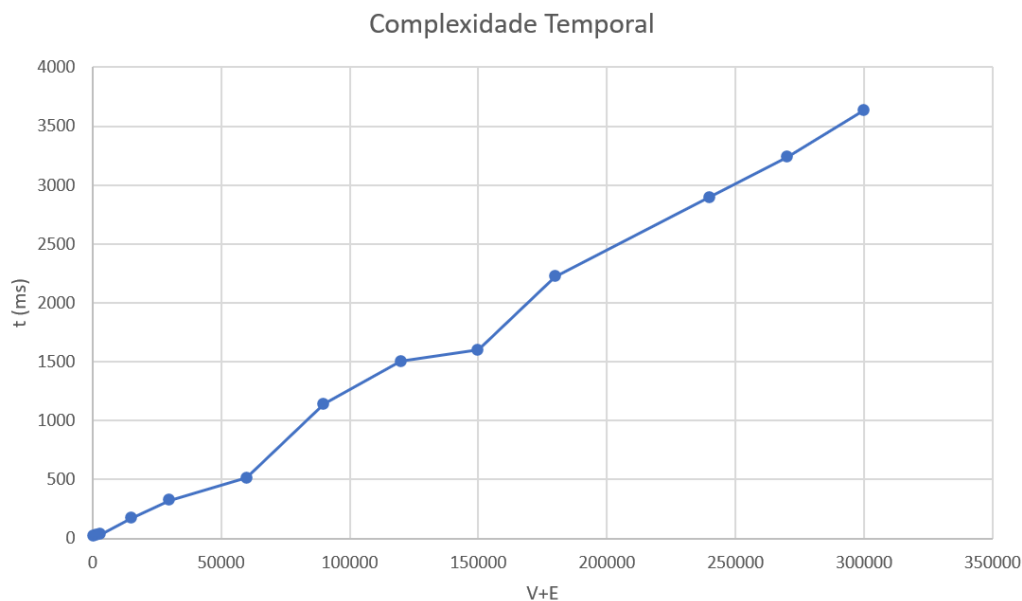
Por fim, de cada BFS retira-se um vetor de nós visitados que representam os antepassados do nó onde a pesquisa começou e é feita uma análise em $O(V)$, percorrendo os nós todos e verificando se são antepassados comuns mais próximos.

É de notar que enquanto se aplica a segunda BFS, o vetor de nós visitados é contruído em função do vetor anterior, acabando logo com os antepassados comuns em vez de os juntar mais tarde.

Em suma, o problema 1 tem complexidade temporal

$$O(V + E + V + E + V) = O(3V + 2E) = O(V + E)$$

Avaliação Experimental dos Resultados



Problema 1

Referências:

<https://www.baeldung.com/cs/lowest-common-ancestor-acyclic-graph>

<https://www.geeksforgeeks.org/detect-cycle-in-a-graph/>

<https://www.geeksforgeeks.org/graph-and-its-representations/>