Análise e Síntese de Algoritmos 2015/2016

1º Projeto - Relatório

Introdução

A tarefa deste projeto consiste em identificar quais as pessoas fundamentais de uma rede. Uma pessoa p é considerada fundamental se o único caminho para a partilha de informação entre outras duas pessoas r e s passa necessariamente por p (sendo p diferente de r e s).

Sabe-se ainda que, existindo uma ligação de partilha entre duas pessoas, a informação pode ser transmitida em ambos os sentidos e que existe sempre uma forma de partilha de informação entre qualquer par de pessoas.

Descrição da solução

Com base nos dados do problema, podemos começar por representar a rede de pessoas como um grafo não dirigido caracterizado por ter só uma componente ligada porque a informação é partilhada nos dois sentidos e é possível transmitir informação entre qualquer par de pessoas.

Neste contexto, verificamos rapidamente que a solução do problema consiste em identificar os vértices do grafo que, se forem removidos, aumentam o número de componentes ligadas do grafo. Estes vértices são conhecidos como "articulation points" ou "cut vertices".

A solução mais simples para este problema seria remover um vértice de cada vez e aplicar a DFS ao grafo resultante para contabilizar o número de componentes ligadas: se o número de componentes aumentasse, então tínhamos removido uma pessoa fundamental. No entanto, para V vértices e E arcos, a complexidade desta solução é quadrática - $O(V^2 + VE)$ - o que é indesejável.

Uma solução alternativa de complexidade linear – O(V+E) - consiste em fazer uma única passagem da DFS no grafo e guardar informações especificas que nos permitem concluir se um determinado vértice do grafo é fundamental.

O algoritmo desta solução aplica a DFS ao grafo e verifica se uma das seguintes condições é verdadeira para cada vértice u da árvore resultante:

1. Se u é a raiz da árvore e tem pelo menos dois filhos;
2. Se u não é a raiz da árvore e tem um filho v tal que nenhum vértice da subárvore gerada por v tem um arco para trás (back edge) com algum dos predecessores de u.

Para verificar o primeiro caso, basta guardar o número de filhos de cada vértice e qual o predecessor de cada vértice: a raiz da árvore DFS é o que tiver o predecessor NIL e será uma pessoa fundamental se tiver pelo menos 2 filhos.

O segundo caso pode ser identificado guardando o tempo de descoberta de cada vértice e o seu tempo de descoberta mínimo , ou seja, qual o vértice com tempo de descoberta mais baixo possível de aceder a partir de uma subárvore gerada por um vértice u.

Análise Teórica

A análise teórica da solução consiste em descobrir a sua complexidade com base na complexidade das suas componentes. Nesta análise, o número de vértices do grafo é representado pela letra V e o número de arcos pela letra E.

A leitura da primeira linha de input (número de vértices e número de arcos) ocorre em tempo de constante – O(1).

A inicialização do grafo corresponde a um ciclo que lê do input os arcos do grafo e os adiciona à lista de adjacências em tempo constante. O tempo de execução deste ciclo depende do número de arcos, ou seja, é O(E).

De seguida, é chamada a função find_fundamentals que é responsável por inicializar os vetores e timer, aplicar a DFS modificada recursivamente e apresentar os pontos fundamentais.

A inicialização dos cinco vetores (visitado, fundamental, descoberta, minimo e parent) de tamanho V+1 e da variável timer ocorre em tempo constante – O(1).

A aplicação da DFS modificada tem complexidade O(V+E) porque todos os vértices e arcos do grafo são percorridos uma única vez.

Finalmente, a apresentação dos pontos fundamentais no output é um ciclo for que percorre todos os vértices em O(V) e todas as comparações para descobrir os máximo e mínimos ocorrem em tempo constate – O(1).

Analisando todas as componentes da solução, podemos concluir que a sua complexidade teórica é O(V+E).

Avaliação Experimental

Na avaliação experimental da solução, foi utilizado um gerador aleatório de grafos com as caraterísticas pretendidas para o problema (não-dirigido e com apenas uma componente ligada) e registados os seguintes resultados.

Teste	Vértices	Arcos	Vértices + Arcos	Fundamentais	Tempo (s)
1	5	5	10	2	<0.01
2	10	10	20	4	<0.01
3	100	100	200	46	<0.01
4	500	500	1000	254	<0.01
5	500	1000	1500	36	0.010
6	1000	1000	2000	525	0.013
7	1000	10000	11000	0	0.034
8	5000	5000	10000	2486	0.026
9	5000	10000	15000	324	0.045
10	5000	15000	20000	55	0.053
11	5000	20000	25000	5	0.068
12	10000	10000	20000	4985	0.045
13	20000	20000	40000	9957	0.064

Recorrendo à análise da tabela de resultados, podemos verificar que o tempo de execução tem um crescimento linearmente proporcional ao número total de vértices e arcos, o que já tínhamos previsto com base na análise teórica da complexidade da solução - O(V+E).

Para além disso, podemos observar que pode existir uma relação entre o rácio de número de vértices/arcos com o número de pontos fundamentais encontrados. Comparando os testes 8 a 11, concluímos que mantendo o número de vértices constante e aumentando o número de arcos, o número de pontos fundamentais diminui bastante. Aliás, no teste 7, o número de arcos é 10 vezes superior ao número de vértices, e não foi possível encontrar um ponto fundamental nesse grafo.

Referências

http://algs4.cs.princeton.edu/41graph/

https://www.cs.purdue.edu/homes/ayg/CS251/slides/chap9d.pdf

http://stackoverflow.com/questions/15873153/explanation-of-algorithm-for-finding-articulation-points-or-cut-vertices-of-a-gr

https://mayanknatani.wordpress.com/2013/06/01/articulation-points-a-k-a-cut-vertices/

http://www.eecs.wsu.edu/~holder/courses/CptS223/spr08/slides/graphapps.pdf