Análise e Síntese de Algoritmos

Relatório do 1º Projeto

(Reconhecimento de Grupos de Partilha)

Grupo 81:

André Silva – 68707

Miguel Faria – 73092

Introdução ao Problema

Tendo como motivação a partilha de informação entre grupos de pessoas e os algoritmos de identificação de componentes fortemente ligados lecionados na disciplina, o projeto consiste na implementação de um programa que, recebendo como entrada pessoas e as suas partilhas, se pretende que identifique:

1. O número de grupos máximos que partilham informação (componentes fortemente ligados);
2. Dos grupos identificados em *1.*, o tamanho do maior;
3. Dos grupos identificados em *1.*, o número de grupos que apenas partilham informação dentro do seu grupo (isto inclui grupos compostos por uma pessoa apenas).

Descrição da Solução Encontrada

A solução deste problema, de reconhecer grupos de partilha, passou por identificar o que era exatamente um grupo de partilha e como identificar um grupo de partilha algoritmicamente.

A primeira parte da resolução do problema foi, então, identificar as características de um grupo de partilha. Após analisar alguns casos de exemplo, conseguimos concluir que um grupo de partilha era identificado por um conjunto de utilizadores que, percorrendo o caminho definido pelas suas partilhas, se conseguia voltar ao utilizador original. Isto é, se tivermos um conjunto de utilizadores {1,2,3,4} e

o conjunto de partilhas {(1,3), (3,4), (4,1), (4,2)}, temos os grupos de partilha {(1,3,4), (2)}, pois se se seguir as partilhas do utilizador 1, 3 ou 4, consegue-se sempre voltar a ele mesmo; no caso do utilizador 2, como este não tem partilhas, então considera-se que constitui um grupo de partilha por si próprio.

Após concluirmos, teoricamente, o que era um grupo de partilha, focámo-nos em como os identificar algoritmicamente. Para tal, baseámo-nos em teoria de grafos e verificámos que um grupo de partilha tem propriedades em comum com um componente fortemente ligado (*strongly connected component* – *SCC*) num grafo. Isto pois num *SCC* o conjunto de nós forma um ciclo no grafo – ou seja, considerando um conjunto de nós num grafo e os arcos que os ligam, pode-se iniciar uma procura num desses nós e eventualmente regressa-se a esse mesmo nó. Assim sendo, após termos verificado este paralelismo entre um *SCC* e um grupo de partilha, decidimos que o melhor algoritmo para identificar um grupo de partilha era o Algoritmo de Tarjan, tendo sido esse o aplicado neste projeto.

O projeto foi implementado em C++.

Análise Teórica da Solução

Como referido na secção anterior, a solução para o problema proposto usou um algoritmo de identificação de componentes fortemente ligados – o Algoritmo de Tarjan.

A escolha deste algoritmo baseou-se em alguns critérios: ser um algoritmo de simples implementação, ser um algoritmo eficiente e ser um algoritmo que ocupe um mínimo de memória possível no computador.

O Tarjan é um algoritmo eficiente pois tem uma complexidade O(V+E), ou seja, o seu tempo de execução depende do número de vértices e de arcos do grafo. Isto torna o algoritmo linear com o tamanho do grafo, já que analisa a lista de adjacências de cada nó e percorre a árvore uma única vez. Deste modo, a chamada recursiva na realização da DFS ocorre apenas uma vez por nó, sendo por isso um algoritmo que, em termos de tempo, é rápido já que recorre sobretudo a acessos diretos a estruturas em memória em vez de procuras sequenciais.

Em termos de memória, o algoritmo ocupa pouca memória pois a única estrutura criada pelo algoritmo é uma pilha onde vão sendo introduzidos os identificadores dos nós já visitados. De resto, todas as estruturas usadas para identificar os *SCC* já existem em memória antes de o algoritmo ser executado.

Após identificar os grupos de partilha, tivemos de identificar os grupos que apenas partilhavam dentro do grupo. Para isso concebemos um algoritmo que percorre as partilhas realizadas por cada utilizador e verifica se realiza partilha com algum utilizador pertencente a outro grupo, devolvendo *verdadeiro* se apenas partilhar com os membros do seu grupo e *falso* caso contrário. Ao processar cada grupo, só percorre a lista de adjacências de um nó se o número de partilhas não for superior à dimensão (número de nós do grupo). Depois de se processar cada grupo, se for identificado que o grupo apenas tem partilha interna, incrementa-se uma variável que regista o número de grupos com esta característica.

Este algoritmo, devido à propriedade de só percorrer a lista de adjacências se esta não for superior à dimensão do grupo de partilha, é linear no número de nós do grupo e basta um desses nós ter um número de partilhas superior ao tamanho do grupo para o algoritmo terminar e retornar *falso*. Isto torna o algoritmo bastante eficiente e capaz de retornar respostas com relativa rapidez para grupos de elevadas dimensões.

Avaliação Experimental dos Resultados

Depois de realizarmos testes exaustivos ao programa que criámos conseguimos concluir que mesmo para casos onde temos 100 000 utilizadores e 95 000 partilhas, ou seja, para um grupo elevado de utilizadores e com um número elevado de partilhas, o algoritmo completa-se, no pior caso, em 0,09s. Isto é, demora 9 centésimos de segundo – o que é muito rápido considerando o universo a que se aplicou o programa.

Seguidamente apresentamos alguns dos testes por nós efetuados de forma a mostrar os resultados obtidos para certos conjuntos de parâmetros de entrada.

Teste	Input	Output	Output Esperado	Tempo de Execução (s)
1	6 8 1 2 2 4 2 5 4 1 4 5 5 6 6 5 6 3	3 3 1	3 3 1	0
2	100000 0	100000 1 100000	100000 1 100000	0.03
3	100000 5 15576 65889 39466 67444 43104 12151 86400 52351 92129 47568	100000 1 99995	100000 1 99995	0.03

Conforme se pode verificar, os resultados para o teste 2 e 3 mostram uma falta de variação no tempo de execução (embora um tenha 5 partilhas e o outro não tenha nenhuma). Isto deve-se ao facto de o número de nós ser muito superior ao número de partilhas, logo vai ser o tempo de percorrer os nós que vai dominar o tempo de execução.

Finalmente, submetemos o nosso projeto à bateria de testes automáticos, na qual passámos a todos os testes dentro do tempo limite.