Análise e Síntese de Algoritmos
Cálculo do Número de Erdős

Lidia Freitas e Jorge Pessoa

22 de Março de 2015

1 Introdução

1.1 O Problema

Com o avanço científico da sociedade o número de artigos publicados aumentou significativamente. Devido a este crescimento e à necessidade crescente de cooperação em busca de resultados cientificos mais elaborados tornou-se popular o cálculo do número de Paul Erdős (um dos maiores matemáticos do século XX).

Este número consiste na distância a que cada co-autor se encontra do colaborador principal, Erdös, sendo que esta distância é acrescida em 1 cada vez que existe uma co-autoria entre dois colaboradores sendo assim nula a distância de Erdös a ele próprio.

1.2 O Input

O ficheiro de input contém informação sobre as colaborações entre cientistas. Como uma colaboração significa dois co-autores terem pelo menos um artigo científico em co-autoria, podem-se obter inputs de co-autorias iguais.

O input é fornecido ao programa da seguinte forma:

- Na 1ª linha encontra-se o número de colaboradores e o número de relações de co-autoria;
- Na 2ª linha encontra-se o número correspondente ao colaborador principal [identificador de Erdös];
- Nas restantes linhas encontram-se as relações de co-autoria entre dois colaboradores, u e v - números inteiros identificadores separados por um espaço em branco.

1.3 O Output

O ficheiro de output consiste em:

- Na 1ª linha encontra-se o valor da maior distância ao colaborador principal, que consideramos M.
- Seguem-se M linhas em que a cada linha i ($1 \leq i \leq M$) se encontra um inteiro correspondente ao número de pessoas com o número de Erdös igual a i.

2 Descrição da Solução

2.1 Linguagem de Programação

A linguagem escolhida para a implementação do projecto foi C++. Esta escolha deveu-se à grande diversidade de estruturas de dados já implementadas e ao facto de permitir um controlo preciso de gestão de memória. Acresce também o factor de ser uma linguagem rápida, o que facilita o desenvolvimento de um algoritmo que supere as restrições temporais impostas.

2.2 Estruturas de Dados

std::list - escolhido porque possui tempos de leitura e de remoção constantes para o inicio e para o final da lista.

std::vector - escolhido porque aloca memória continua e permite tempos de acesso constantes a qualquer posição do mesmo.

2.3 A Solução

Primeiramente foi definida a representação do domínio do problema para chegar a uma solução: cada autor foi representado por um vertice, cada colaboração por uma ligação entre vértices, sendo assim o problema transposto para um grafo não dirigido, onde o número de Erdös para cada autor seria apenas a distância do vértice que o representa ao vértice que representa Erdös.

Desta forma o problema reduz-se ao cálculo de distâncias dentro de um grafo permitindo usar um dos vários algoritmos estudados, o BFS.

2.4 O Algoritmo

Após ser criado o grafo que representa o input (lido do stdin) é necessário calcular a distância de cada vértice ao vértice de Erdös. Para tal é executado o algoritmo de BFS a partir do vértice de Erdös no grafo, calculando assim as distâncias de cada vértice da forma que se segue:

- Todos os vértices são inicializados a branco (indicando que nunca foram tratados pelo algoritmo) e com distância 'infinita'.
- Começando do vértice de Erdös, é lhe atribuido a distância de 0 e é colocado na fila de espera dos vértices a serem visitados.
- Quando um vértice v é visitado são percorridos todos os seus vértices adjacentes incrementando a distância em 1 relativo a v Estes são adicionados à fila de espera para serem visitados, e o pai é retirado desta fila.
- Quando já não existem vértices na fila o algoritmo termina.

Em comparação a uma implementação geral do BFS, procedemos à remoção da cor preta e das referências aos vértices predecessores por falta de utilidade na implementação necessária para o projeto e de forma a melhorar a eficiência deste. Os vértices podem assim estar brancos (não tratados pelo algoritmo) ou cinzentos (já adicionados à lista de vértices a visitar), sendo que a sua remoção da lista de visitados é equivalente à mudança para cor preta.

À medida que são percorridos os vértices adjacentes estes são removidos para facilidade de uso, visto que o algoritmo apenas percorre cada vértice adjacente de outro vértice uma vez. A 'destruição' da lista de adjacências de um vértice não tem assim qualquer implicação para o algoritmo em si.

3 Análise Teórica

4 Avaliação Experimental dos Resultados

5 Bibliografia

1. Enunciado do Projecto
2. Slides e Apontamentos das Aulas Teóricas