Relatório Projecto ASA

Primeira Parte

TÉCNICO LISBOA

Grupo 64:
Ricardo Costa Dias Rei, 78047
Mariana Gaspar Fernandes, 76448

Breve Introdução:

Como problema foi-nos pedido para calcular o numero de Erdõs cientistas. Este problema consiste em determinar a distância colaborativa de um conjunto de cientistas a Paul Erdõs. Considera-se que os co-autores de artigos com Paul Erdõs estão à distância de 1. Os co-autores de artigos com co-autores de Paul Erdõs estão à distância de 2, e assim sucessivamente.

O objectivo do trabalho era desenvolver um programa que ao receber o número de cientistas (em que cada cientista é representado por um inteiro positivo), o número que representa o Paul Erdõs, e as diversas colaborações.

Descrição da solução:

Ao olhar para o problema identificámos que poderia ser traduzido num grafo em que cada pessoa representa um vértice e as co-autorias representam arestas.

Assim sendo, criámos uma função buildGraph que ao receber o input construía o grafo correspondente. Na construção do grafo é utilizada a estrutura Graph que tem o campo "_paul" para guardar o vértice que representa o Paul, o campo "verticesNumber" que guarda o numero de vértices, o campo "_vertices" para o vários vertice e o campo "adjacentList" que representa as arestas. A utilização de uma lista de adjacências pareceu-nos mais correcta uma vez que os grafos são maioritariamente esparsos.

Depois aplicando o algoritmo breadth-first-search ao grafo vamos conseguir saber a distância (mais curta) de cada vértice (co-autor) ao vértice que representa o Paul. Como suporte à execução do mesmo, a estrutura Vertex tem o campo "d" para guardar a distância ao Paul, o campo "pi" que é um ponteiro para o predecessor e o campo color que vai ser usado para saber se o vértice já foi ou não visitado. A escolha deste algoritmo foi a que nos pareceu mais adequada com base na matéria dada nas aulas teóricas.

Por fim usámos uma função a que chamamos outPut que procura o valor máximo da distância ao Paul e cria um vector com esse tamanho esse vector é depois utilizado como contador para as distâncias ao mesmo (estilo counting sort).

Análise Teórica de Resultados:

O nosso projecto tem como limite assimptótico superior O(V+E+M) em que V representa o número de vértices, E é o número de arestas e M a distância máxima que um co-autor tem a Paul. As funções mais importantes a nível de complexidade no programa são:

- buildGraph que percorre todos os vértices para os inicializar e todas as arestas para construir a lista de adjacências O(V+E).

- breadthFirstSearch percorre todos os vértices com caminhos até ao vértice do Paul e para cada vértice todas as suas arestas existentes na lista de adjacências. Uma vez que os vértices vão sendo marcados com cores eles apenas são percorridos uma vez mesmo que aparecendo várias vezes na lista de adjacências O(V+E).

- Por fim a função outPut que percorre todos os vértice à procura da distância máxima ao Paul, e cria um vector com esse tamanho que mais tarde é percorrido para gerar o output como explicado anteriormente O(V+M).

Análise experimental:

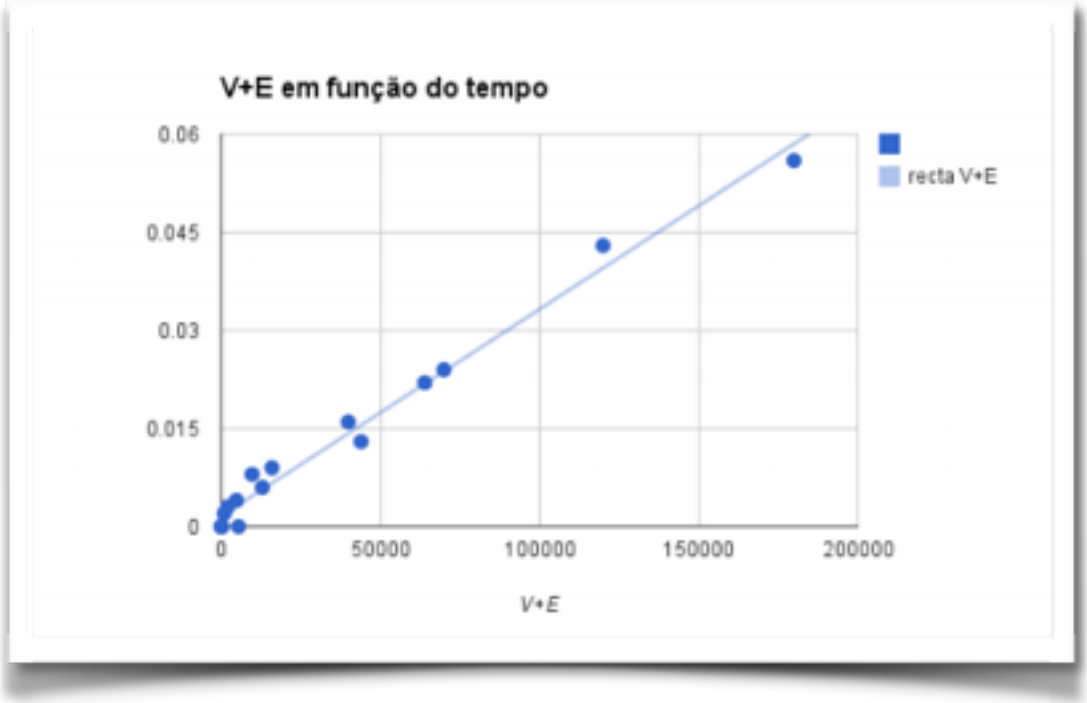

Grafico nº1

O gráfico nº1 é o resultado de alguns testes realizados em que o pretendido era verificar a relação entre o tamanho do grafo e o tempo que o algoritmo demora. Para isso foram medidos os tempos para inputs com grafos de diferentes tamanhos. O tamanho de um grafo é dado pela função V + E. Nos testes realizados para a construção deste gráfico o valor de M é constante (distância máxima a Paul é 10).

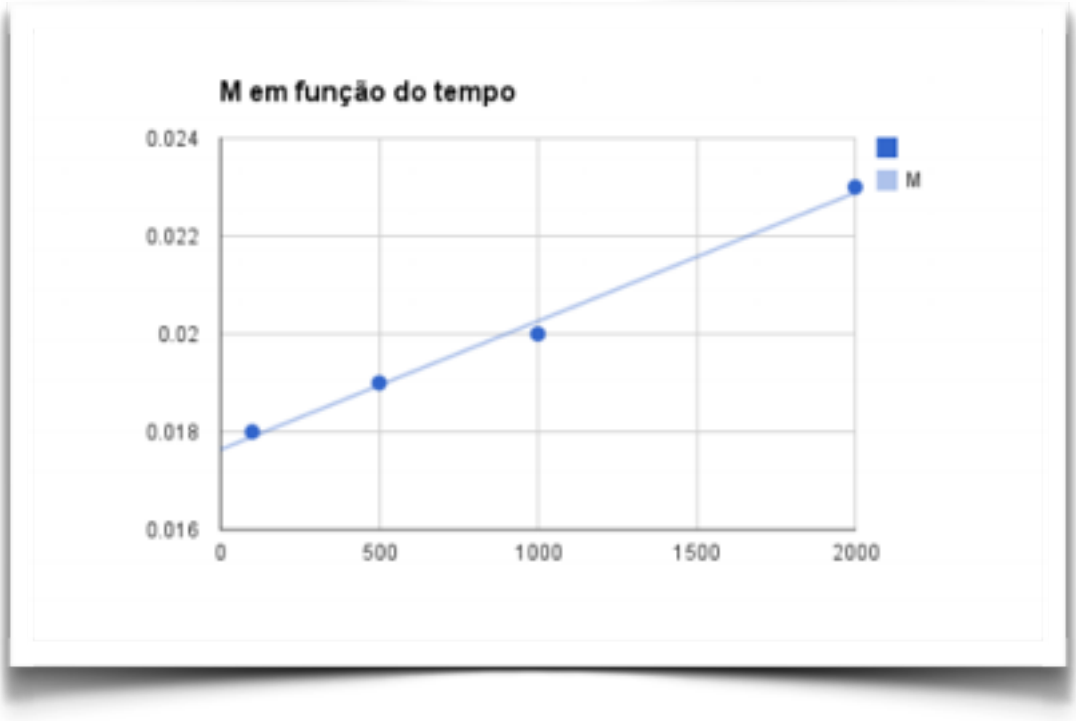

Grafico nº 2

O gráfico nº 2 é o resultado dos testes realizados para medir a influência do M, sendo M o valor máximo da distância de um co-autor a Paul. Apesar de este valor não ser tão influente como o número de vértices ou arestas achámos interessante realizar uma análise experimental simples com apenas 4 testes. Para esta análise usámos um grafo com 22 000 vertice e 21 999 arestas e diferentes valores de M. O resultado é apresentado em cima.

Confirmamos portanto com base no gráfico 1 e no gráfico 2 que o nosso algoritmo é linear em função de V, E e M, O(V+E+M) como apresentado na analise teórica.