**Análise e Síntese de Algoritmos (2º Projeto)**

Grupo 57

Isabel Soares (89466)

Tiago Afonso (89546)

1. **Breve introdução**

Foi-nos proposto desenvolver um projeto no âmbito da cadeira de Análise e Síntese de Algoritmos que se baseia numa aplicação informática que determina a capacidade máxima que a sua rede é capaz de transportar.

Deste modo, apresentamos o problema como um grafo dirigido e pesado onde utilizamos um algoritmo de pré-fluxos para calcular o fluxo máximo nomeadamente Relabel-To-Front lecionado nas aulas.

1. **Descrição da solução**

A implementação do programa foi realizada em linguagem C.

Inicialmente, começamos por implementar o algoritmo Relabel-To-Front. Através de funções auxiliares tais como ‘push’,‘relabel’ , ‘discharge’, ‘moveToFront’, ‘pushRelabel’ calculamos a capacidade máxima da rede.

De seguida, devido à utilização de duas matrizes: uma para guardar a informação dos fluxos e outra que contém a informação referente a cada uma das capacidades, encontramos a rede residual através da subtração de ambos os valores da matriz de fluxos com os valores da matriz das capacidades e transpusemos a matriz resultado dessa subtração.

Implementámos também uma DFS para encontrar o corte mínimo do grafo anterior.

1. **Análise Teórica**

Em relação à análise teórica do nosso algoritmo, mais concretamente relacionado com a execução de cada ciclo, tendo em conta que V é o número de vértices de um grafo (neste caso os fornecedores(f), as estações de abastecimento(e) e o Hipermercado) e E é o número de arestas de um grafo (neste caso o número de ligações que existem na rede do Sr. Caracol), temos:

* Algoritmo Relabel-To-Front: O(V3);
* Matriz de Adjacências: O(V2);
* Algoritmo DFS:O(V + E);

Logo, a complexidade final do nosso algoritmo é de O().

1. **Análise Experimental dos Resultados**

Para verificar o uso de memória e tempo de execução do algoritmo, foram testados os inputs dados como exemplo, utilizando as funções *time* e *Valgrind* do Linux.

Como se pode ver, o gasto de memória é linear, aumentando com o número de fornecedores, estações de abastecimento e ligações, não sendo tão fácil de ver no gráfico Tempo-Routers+Ligações, devido à diferença de valores usados nos exemplos e à diferença de precisão entre o comando *time* e o *Valgrind*

1. **Referências**

* **Introduction to Algorithms, Third Edition:**Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein September 2009 ISBN-10: 0-262-53305-7; ISBN-13: 978-0-262-53305-8
* <https://stackoverflow.com/questions/4482986/how-can-i-find-the-minimum-cut-on-a-graph-using-a-maximum-flow-algorithm>