

Análise e Síntese de Algoritmos

Relatório do 2º projecto

Grupo 96

1 Introdução

O 2º projecto da cadeira de Análise e Síntese de Algoritmos do ano curricular 2018/2019 consiste na identificação da capacidade máxima de uma rede. Tem ainda como objectivo identificar as estações de abastecimento e as ligações que devem ser aumentadas.

O **input** descreve a rede de distribuição. O input é então constituído por:

- uma linha com o número de fornecedores($f > 0$), de estações de abastecimento($e \geq 0$) e de ligações($t \geq 0$) que existem na rede(separados por um espaço).
- uma linha com **f inteiros**, separados por um espaço, que representam a produção de cada fornecedor
- uma linha com **e inteiros**, separados por um espaço, que representam a capacidade de cada estação de abastecimento
- uma sequência de **t linhas** com 3 inteiros: a origem da ligação($o \geq 2$), o destino da ligação($d \geq 1$) e a capacidade da ligação($c \geq 1$).

Para a identificação de uma ligação sabemos que o Hiper é sempre o número 1 e que os números de 2 a $f+1$ representam fornecedores.

No **output** uma sub-rede é identificada pelo router com maior identificador que pertence à sub-rede. É constituído por:

- uma linha com a capacidade máxima da rede
- uma linha com os números das estações de abastecimento que devem ser aumentadas, separadas por um espaço e por ordem crescente. Caso não existam estações a ser aumentadas, deve ser apresentada uma linha apenas com o caracter de fim de linha.
- uma sequência de linhas com os valores de origem e de destino das ligações que devem ser aumentadas sendo que devem ser reportadas apenas aquelas que definem a capacidade da rede e dentro destas apenas as mais perto do Hiper. Esta sequência deve ser ordenada por ordem crescente, considerando primeiro a origem e apenas em caso de empate o destino. Caso não existam ligações a ser aumentadas, nada deve ser apresentado.

2 Descrição da Solução

2.1 Linguagem de programação

A implementação do programa foi elaborada na linguagem de programação C++.

2.2 Estruturas de dados

Para a implementação do algoritmo recorreremos a **duas estruturas**, uma que representa os vértices(Vertex) e outra que representa uma ligação entre dois vértices(Edge). Utilizamos **dois vetores de vértices** que após serem unidos irão representar a rede de distribuição. Para além disso temos uma **fifo/queue** utilizada durante a aplicação do Push Relabel. Recorreremos ainda a um **vetor auxiliar** utilizado para ordenar as ligações de forma a que sejam impressas como pedido no enunciado.

2.3 Solução

Para a resolução do problema decidimos utilizar uma rede de fluxo em que os fornecedores, as estações de abastecimento, a source e o hiper(sink) se tratam dos vértices e as ligações(incluindo as criadas desde a source até todos os fornecedores) são as arestas(?) desta rede de fluxo. Foi identificado que a solução podia ser resolvida transpondo a rede ao receber o input(de forma a facilitar encontrar as estações de abastecimento e as ligações que devem ser aumentadas) e utilizando o algoritmo Push Relabel(que permite encontrar a capacidade máxima da rede) com uma queue/fifo para acelerar o processo. METO AQUI A CENA DAS ALTURAS??

2.4 Algoritmo

FAZER

3 Análise

3.1 Análise Teórica

FAZER

3.2 Análise Experimental

Para verificar a linearidade do tempo de execução do projecto e para uma melhor análise, recorreremos ao gerador de grafos fornecido pelos docentes.

ACABAR

4 Referências

Os websites/obras consultados para realizar o projecto foram os seguintes:

- https://en.wikipedia.org/wiki/Push%E2%80%93relabel_maximum_flow_algorithm
- <https://fenix.tecnico.ulisboa.pt/downloadFile/1970943312332951/ch26a.pdf>
- **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