

Instituto Superior Técnico
Análise e Síntese de Algoritmos
Relatório 1º Projeto

Grupo 94 (Alameda) | João Palet - 86447 | Maria Forjó - 86475

Introdução

O problema proposto baseia-se em dividir a rede de supermercados do Sr. João Caracol em sub-redes, de modo a que este consiga ter sub-redes regionais de distribuição na sua cadeia, para que numa região seja possível transportar produtos entre quaisquer dois pontos.

O objetivo do projeto é desenvolver um método eficiente de obter o número de sub-regiões e quais as ligações entre as mesmas. O sistema desenvolvido recebe do *standard input* os números de supermercados existentes na cadeia e de ligações existentes entre os mesmos, tal como as respetivas ligações, e apresenta para o *standard output* a resolução do problema.

Descrição da Solução

Depois de analisado o problema, concluiu-se que a melhor forma de o resolver passaria pela implementação de um grafo dirigido, onde cada supermercado da cadeia do Sr. João seria representado por um vértice, e cada rota de transporte por uma aresta nesse mesmo grafo.

A solução implementada consiste em ler do *standard input* o número de supermercados da cadeia ($N \geq 2$), seguido do número de ligações ($M \geq 1$) entre eles e ainda de uma lista de M linhas com todas essas ligações, em que cada linha contém 2 inteiros separados por um espaço ($u\ v$), significando que existe uma rota direta de distribuição de u para v . Dadas estas garantias, não se considerou necessário realizar a verificação do *input*.

Deste modo, o grafo foi implementado com recurso a uma lista de adjacências. Ao ser lida do *standard input* uma ligação entre dois vértices é

estabelecida a respetiva correspondência na lista de adjacências, onde no início da lista, no índice correspondente ao primeiro vértice, é inserido um nó contendo o segundo vértice, e assim sucessivamente. À medida que isto é feito, vai sendo construído um array com todas as ligações do grafo. Neste array, cada ligação é representada como uma estrutura que contém dois inteiros, o vértice de partida e o de chegada.

Terminadas a inicialização e construção das estruturas escolhidas, procede-se à identificação das componentes fortemente ligadas do grafo, ou seja, as sub-redes regionais. Optou-se pelo algoritmo de Tarjan pois embora assintoticamente seja semelhante ao de Kosaraju, tem um fator constante muito menor. Foram então criados uma stack e quatro arrays auxiliares, onde a informação do vértice x está presente na posição de índice x . Os três arrays são:

- d - para os tempos de descoberta;
- low - para os valores de low;
- on_stack - para ser possível saber em tempo constante se um certo vértice está na stack;
- SCC - para associar a cada vértice a componente a que pertence.

Findo o algoritmo de Tarjan, cada índice do vetor SCC tem um número que o associa à sua respetiva componente (ex: as posições cujos índices são os vértices da primeira componente encontrada têm 0s, as com os índices dos vértices da segunda componente encontrada têm 1s, etc). Para transformar estes números no número do vértice com menor identificador que pertence à componente, basta percorrer o vetor e a primeira vez que se encontra cada número, o valor do índice onde isso acontece é o representante da componente.

Quando cada vértice já está associado ao representante da componente a que pertence, transforma-se o array de ligações entre vértices num array de ligações entre componentes. A partir daqui, basta retirar as ligações entre uma componente e ela própria, ordenar o array de forma crescente e retirar as ligações repetidas.

No *standard output* são apresentados o número de componentes fortemente ligadas, o número de ligações entre elas e uma sequência ordenada dessas mesmas ligações.

Análise Teórica

Para ser analisada a complexidade do algoritmo implementado na solução é feita a análise das secções que o constituem.

Na sua totalidade, a inicialização e construção do grafo é $O(V+E)$, sendo V o número de vértices e E o número de arcos existentes no grafo. Na construção, a inserção das ligações entre vértices na lista de adjacências é efetuada em tempo constante $O(1)$. A execução do algoritmo Tarjan ocorre em tempo linear, $O(V+E)$, pois a inicialização é $O(V)$ e a análise dos arcos na função *tarjan_visit* ocorre em $\Theta(E)$. Encontrar o representante da SCC correto para os vértices de cada componente dá-se em $O(V)$, pois são percorridos todos os vértices. Para ordenar o vetor com as ligações entre as componentes, é aplicado o algoritmo de ordenação quicksort, que tem complexidade $O(E * \log E)$ no caso médio, o que poderá influenciar a complexidade do algoritmo. Por fim, imprimir para o *standard output* as ligações entre supermercados ordenada é $O(V)$.

Conclui-se então com base na complexidade das diferentes secções que o programa da solução desenvolvida é executado aproximadamente em tempo linear, $O(V+E)$ excepto quando o número de arestas a ordenar é tão elevado que passa a ser ele a ditar a complexidade do algoritmo, passando esta nesse caso a ser $O(E * \log E)$.

Avaliação Experimental dos Resultados

Para analisar o programa, foram realizados vários testes recorrendo ao gerador facultado. Foi aumentado progressivamente o tamanho do *input* de modo a estudar o crescimento do tempo de execução do programa. Foi utilizado o comando *time*, de forma a extrair o *real time* quatro vezes, tendo sido calculada a média aritmética entre estes para obter os valores apresentados no gráfico.

O tempo de execução obtido é aproximadamente linear $O(V + E)$, e como seria de esperar a complexidade da solução desenvolvida foi influenciada pela complexidade do algoritmo de ordenação quicksort $O(E * \log E)$, como previsto na análise teórica. O pequeno desvio experimental deve-se também a fatores externos não controláveis.

V	E	V + E	n° SCC	Tempo de Execução (s)
0	0	0	0	0
4000	6000	10000	500	0.005
8000	12000	20000	1000	0.006
12000	18000	30000	1500	0.008
16000	24000	40000	2000	0.014
20000	30000	50000	2500	0.022
24000	36000	60000	3000	0.028
28000	42000	70000	3500	0.035
32000	48000	80000	4000	0.046

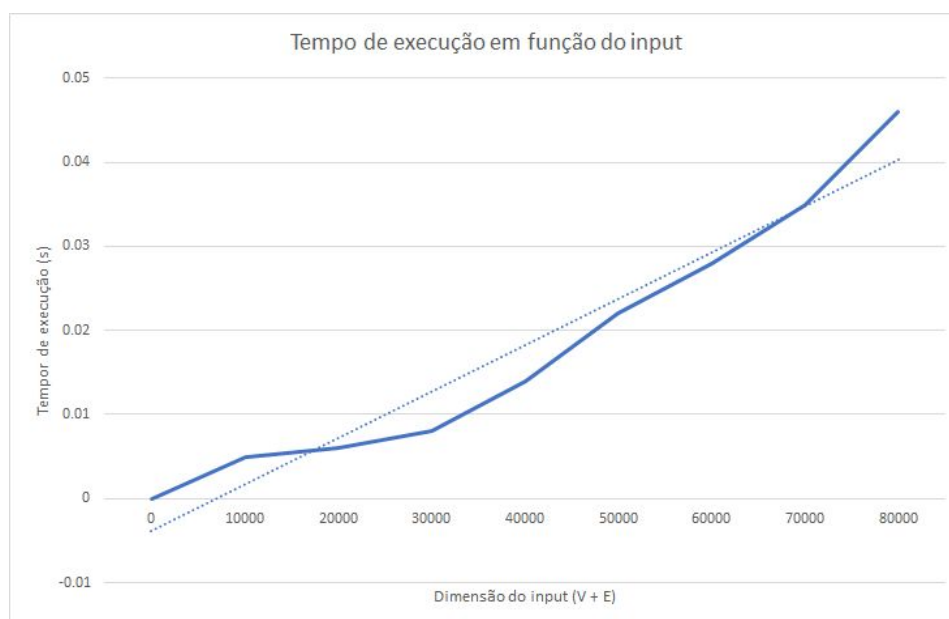


Fig. 1 - Gráfico representativo da análise experimental da solução desenvolvida

Comentários e Referências

Foi escolhida a linguagem C para o desenvolvimento da solução ao problema proposto pela sua eficiência e pelo controlo sobre a memória utilizada e sobre as estruturas necessárias que esta possibilita.

O algoritmo Tarjan utilizado foi baseado no pseudocódigo sugerido nas aulas teóricas de ASA 16/17.