

# Relatório 2º Projeto ASA 2023/2024

**Grupo:** al138

**Nome:** João Miguel Calejo Alcalde Teixeira, 106634

## **Descrição do Problema e da Solução:**

Para a resolução do problema foi feita uma leitura do input, utilizando um ciclo para registar todas as relações entre as pessoas num vetor de vetores. A representação deste problema é feita através de grafos orientados em que as pessoas são representadas por vértices e as relações pelos arcos.

O objetivo é calcular o maior número de saltos entre as componentes fortemente ligadas do grafo, SCC's. Para isto é usado o algoritmo Depth-first search, DFS, no grafo inicial de modo a obter um vetor com a ordem de tempo de fim. Depois é aplicado de novo o algoritmo DFS no grafo transposto de forma a identificar as SCC's há medida que decorre este processo é verificado o máximo de saltos possíveis de dar até chegar á SCC iterada.

Uma dificuldade que o exercício proporcionou foi a aplicação dos algoritmos de forma iterativa.

## **Análise Teórica:**

- A leitura dos dados iniciais que envolve um Loop que itera E vezes, sendo E o numero de relações e constrói dois grafos num vetor de vetores usando push\_back() que tem complexidade  $O(1)$ . Assim a complexidade deste ciclo é  $O(E)$ .
- Depois obteve-se o vetor da ordem, tempo de fim. Envolveu um ciclo que itera V vezes e por cada iteração é realizada a DFS ao vértice iterado, complexidade  $O(V+E)$  , assim a complexidade deste processo é  $O(V(V+E))$  sendo V o número de vértices. Além disso é invertida a ordem do vetor que armazena a ordem pretendida, e tem uma complexidade  $O(V)$  que é de ordem inferior á complexidade antes calculada. Assim este procedimento tem uma complexidade de  $O(V(V + E))$ .
- Por fim para calcular o máximo de saltos, é mais uma vez realizado um ciclo que itera V vezes e executa o algoritmo DFS, complexidade  $O(V + E)$ . Esta chamada da DFS é muito semelhante á anterior, apresentado mais um Loop que itera os vértices contidos na SCC em procura  $O(V)$ , como a complexidade deste ciclo é da mesma ordem, mas menor que a complexidade da DFS então é o ciclo não é relevante para a complexidade final do processo. Deste modo a complexidade final deste procedimento é  $O(V(V + E))$ .
- Conclui se então que a complexidade do programa é  $O(V(V + E))$ .

## **Avaliação Experimental dos Resultados:**

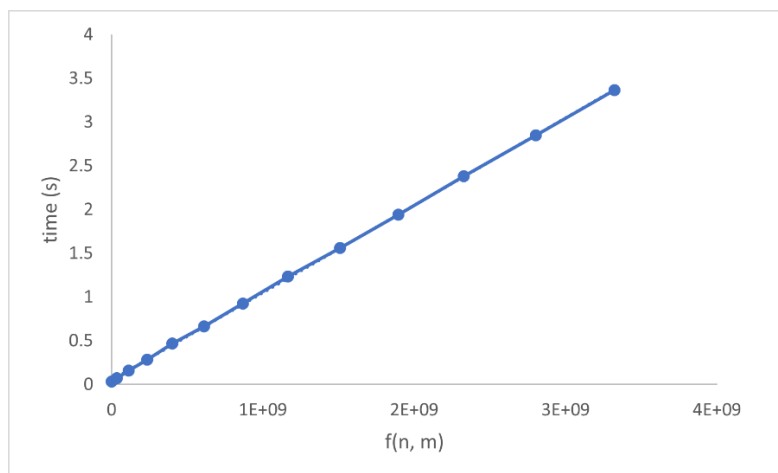
De forma a analisar o algoritmo foi usado um programa, gerador de instâncias, que gerou um input para um determinado numero de indivíduos, relações, sub-redes, minimo e máximo de indivíduos por sub-rede.

Os dados que geraram as instâncias foram registados numa tabela juntamente com o tempo de execução respeticvo de cada instancia.

V	E	SCC	MIN	MAX	time (s)
800	1000	200	1	10	0.032
3600	6100	700	1	10	0.072
6400	11200	1200	1	10	0.16
9200	16300	1700	1	10	0.282
12000	21400	2200	1	10	0.469
14800	26500	2700	1	10	0.663
17600	31600	3200	1	10	0.927
20400	36700	3700	1	10	1.236
23200	41800	4200	1	10	1.562
26000	46900	4700	1	10	1.941
28800	52000	5200	1	10	2.38
31600	57100	5700	1	10	2.847
34400	62200	6200	1	10	3.364

*Figura1- Tabela com dados das instâncias geradas.*

De modo a verificar o resultado da análise teórica, foi feito um novo gráfico com o tempo a variar em função da complexidade prevista. É esperado obter um crescimento linear do tempo em função dos dados da tabela, de forma a verificar a complexidade do problema.



*Figura2- gráfico em função da complexidade da análise teórica,  $f(n,m)=V(V+E)$*

Com uma reta completamente sobreposta à linha de tendência, fica provado que a complexidade do código realizado para resolução do problema proposto é  $O(V(V+E))$ .