

# Relatório 2º Projeto de ASA – 2023/2024

Grupo: AL059

Aluno(s): Joana Vaz (106078) e Martim Afonso (106507)

---

## Descrição do problema e da solução:

O problema abordado neste projeto consiste em encontrar o comprimento máximo de um caminho num grafo direcionado. A solução que encontramos utiliza duas componentes principais: uma passagem em profundidade para encontrar a ordem de saída dos vértices no grafo e uma segunda passagem em profundidade para calcular os comprimentos máximos dos caminhos a partir de cada vértice, considerando as componentes fortemente ligadas. A solução proposta tem por base a utilização de um algoritmo de DFS, que procura encontrar o caminho máximo possível.

## Análise Teórica:

A solução que encontramos está dividida em duas funções principais: `iterativeDFS` e `iterativeDFS2`.

- A função `iterativeDFS` faz uma passagem em profundidade pelo grafo obtido através do input do utilizador para encontrar a ordem de saída dos vértices no grafo. Ao fazer isto, existe uma complexidade:  $O(V + E)$ , onde  $V$  é o número de vértices (pessoas) e  $E$  é o número de arestas (relações) no grafo.

- A função `iterativeDFS2` utiliza a ordem de saída dos vértices para fazer uma passagem em profundidade no grafo transposto e calcular os comprimentos máximos dos caminhos a partir de cada vértice, considerando as componentes fortemente ligadas. Todo este processo tem uma complexidade de  $O(V + E)$ , onde  $V$  é o número de vértices (pessoas) e  $E$  é o número de arestas (relações) no grafo.

- Leitura dos Dados de entrada:

A leitura do input consiste em obter o número de pessoas ( $n$ ) e o número de relações ( $m$ ) entre as pessoas. Um ciclo linear dependente de  $n$  e  $m$ , e, por isso, a complexidade é de  $O(n+m)$ .

- Processamento da instância para fazer alguma coisa:

O processamento identifica o caminho mais longo entre quaisquer duas pessoas na rede. A aplicação de um algoritmo para encontrar caminhos mais longos, como por exemplo um DFS, é uma abordagem eficiente. A complexidade é de  $O(n+m)$ .

- Aplicação do algoritmo indicado para cálculo da função recursiva:

A aplicação do algoritmo para calcular o valor desejado é realizada sobre o grafo da rede TugaNet, que ao utilizar um algoritmo de DFS a complexidade é de  $O(n+m)$ .

- Apresentação de dados:

Na apresentação do resultado, apenas é mostrado o número máximo de saltos que a doença por realizar, e, por isso, é considerado uma operação simples, com complexidade de  $O(1)$ .

## Complexidade global da solução:

A complexidade global da solução é dominada pela aplicação do algoritmo, sendo este  $O(n+m)$ . Esta complexidade é bastante aceitável devido ao tamanho da entrada esperada e dada a natureza do problema.

# Relatório 1º Projeto de ASA – 2023/2024

Grupo: AL059

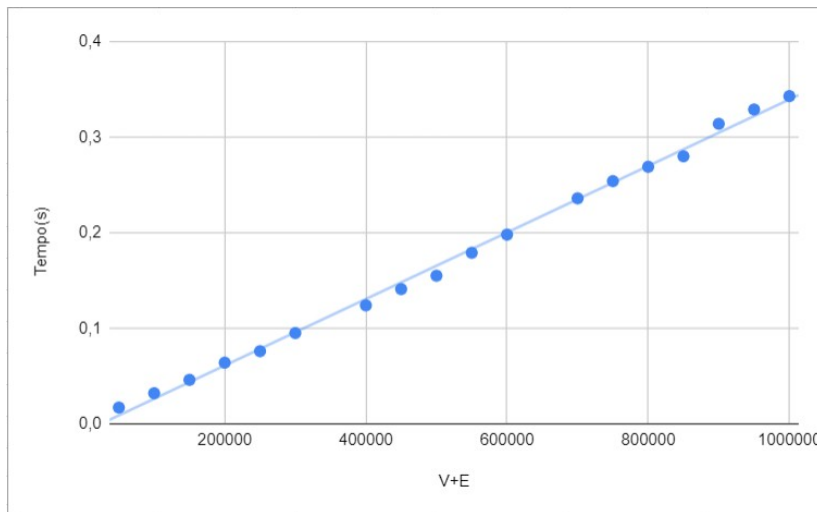
Aluno(s): Joana Vaz (106078) e Martim Afonso (106507)

---

## Avaliação Experimental dos Resultados:

Recorrendo ao gerador, gerámos um total de 18 instâncias com um incremento de 50000 no valor total de  $V+E$  e obtivemos o seguinte gráfico:

esperado pela análise teórica onde  $f(V,E)$  é igual a  $V+E$  e obtivemos o seguinte gráfico:



V	E	T	V+E
28600	21400	0,017	50000
43900	56100	0,032	100000
68000	82000	0,046	150000
107500	92500	0,064	200000
129000	121000	0,076	250000
158000	142000	0,095	300000
200000	200000	0,124	400000
228000	222000	0,141	450000
251000	249000	0,155	500000
280000	270000	0,179	550000
310000	290000	0,198	600000
330000	370000	0,236	700000
340000	410000	0,254	750000
375000	425000	0,269	800000
400000	450000	0,28	850000
415000	485000	0,314	900000
445000	505000	0,329	950000
475000	525000	0,343	1000000

Ao observar o gráfico, podemos verificar que existe uma relação linear entre a complexidade teórica prevista e os tempos registados, confirmando assim que a implementação está de acordo com a análise e previsão teórica.