

# Relatório 2º Projeto ASA 2023/2024

**Grupo:** tp055

**Aluno:** Enzo Nunes ist1106336

## Descrição do Problema e da Solução

- O problema consiste em encontrar o tamanho do maior salto que uma doença pode dar dentro de uma rede de pessoas, considerando que a doença se espalha uma pessoa de cada vez, e infeta instantaneamente ciclos de pessoas que se conhecem todas entre si.
- A solução apresentada traduz o problema num grafo dirigido, sendo o objetivo encontrar o maior caminho possível nesse grafo, considerando cada ciclo como um único vértice.

## Análise Teórica

A solução mais simples para este problema utilizaria procuras em profundidade (DFS) e/ou o algoritmo de *Tarjan* para *SCCs*, por exemplo. De qualquer modo, estes algoritmos naturalmente são recursivos. A solução apresentada apresenta algumas otimizações a estes algoritmos.

O programa começa por montar um grafo, representado por vetores de adjacências, e aplica o algoritmo de *Tarjan* neste grafo.  $O(V + E)$

O algoritmo de *Tarjan* utilizado neste programa para encontrar Componentes Fortemente Conectados (*SCCs*) em um grafo dirigido utiliza uma abordagem de busca em profundidade (*DFS*) de maneira iterativa onde uma *stack* é mantida para substituir as chamadas recursivas. Cada vértice recebe um índice único quando é visitado pela primeira vez e também um *low-link*, que é o menor índice de qualquer vértice alcançável a partir dele, incluindo o próprio nó. O algoritmo explora iterativamente o grafo, e identifica *SCCs* quando um vértice tem como *low-link* o seu próprio índice. Ao encontrar um vértice como este, são separados da *stack* todos os vértices até e incluindo este, formando um *SCC*. Esse processo iterativo continua até que todos os vértices tenham sido visitados.  $O(V + E)$

Em seguida, é montado um grafo dirigido acíclico (*DAG*) onde cada vértice é um *SCC* obtido anteriormente. Considera-se as mesmas arestas do grafo original desde que sejam de *SCCs* diferentes.  $O(V + E)$

Dada a natureza do *Tarjan*, que é baseado em *DFSs*, este *DAG* já vem topologicamente ordenado. Basta então percorrer este grafo saindo do vértice com maior índice e computar a maior distância percorrível no grafo.  $O(V + E)$

# Relatório 2º Projeto ASA 2023/2024

**Grupo:** tp055

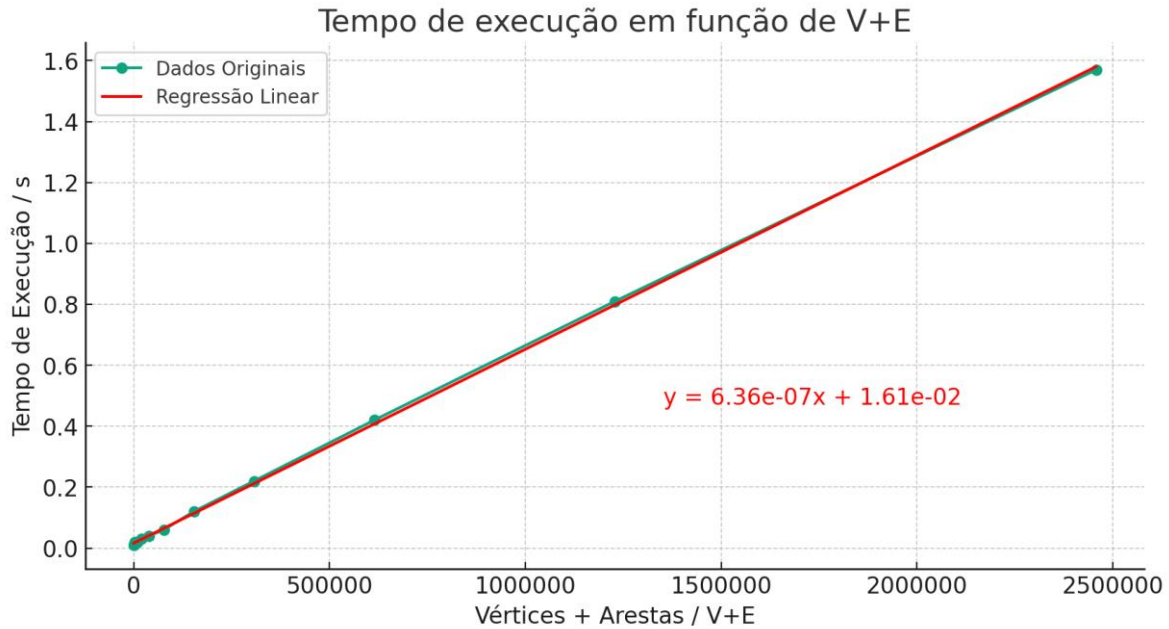
**Aluno:** Enzo Nunes ist1106336

Todos estes passos tem uma complexidade linear a  $O(V + E)$ , pelo que a complexidade final para problemas maiores também vai ser na ordem de  $O(V + E)$ .

## Avaliação Experimental dos Resultados

Foram utilizadas 15 instâncias para testar o tempo de execução do programa. O tempo de execução guardado para cada uma resulta da média aritmética entre 5 testes realizados em cada uma.

V	50	100	200	400	800	1600	3200	6400	12800	25600	51200	102400	204800	409600	819200
E	100	200	400	800	1600	3200	6400	12800	25600	51200	102400	204800	409600	819200	1638400
V+E	150	300	600	1200	2400	4800	9600	19200	38400	76800	153600	307200	614400	1228800	2457600
T/s	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.06	0.12	0.22	0.42	0.81	1.57



O gráfico mostra que claramente existe uma relação linear entre o tempo de execução e a soma do número de vértices e número de arestas, o que confirma que a implementação proposta está de acordo com a análise teórica feita previamente.