Relatório 2º projeto ASA 2023/2024

Grupo: TP049
Aluno(s): Henrique Machado (103202) e Vasco Pereira (103368)

Descrição do Problema e da Solução

Foi nos dado um problema em que tínhamos uma lista de relações direcionadas entre indivíduos e foi nos pedido para encontrar o número máximo de saltos que uma doença podia fazer nesta rede, sendo que indivíduos que se conheçam mutuamente de forma direta ou indireta ficam infetados instantaneamente. Para resolver isto começámos por criar um grafo direcionado G, com V vértices (número de indivíduos) e E arcos (número de relações entre indivíduos) e o seu transposto G^T. De seguida identificamos que se transformássemos o grafo G no grafo G^{SCC} juntaríamos todos os indivíduos que se conhecessem mutuamente de forma direta ou indireta num único vértice. Para fazer isto aplicámos o algoritmo $DFS(G) + DFS(G^T)$ para obter os grupos SCC. Para criar o grafo G^{SCC} visitámos todos os arcos do grafo G e verificámos se para cada arco ambos os indivíduos se encontravam em grupos SCC diferentes, se sim adicionámos um arco entre o grupo SCC de um indivíduo e do outro com a mesma direção. Após termos o grafo G^{SCC} aplicámos o algoritmo DFS nesse grafo para obtermos a ordem de fecho de cada vértice. Após termos a ordem de fecho de cada vértice criámos um vetor *distances*, de tamanho igual ao número de vértices de G^{SCC}, inicializado a -1 para cada valor. A partir disto iteramos por cada vértice v pela ordem inversa de fecho, verificamos se $distances[v] == -1$ se sim então $distances[v] = 0$ e verificamos para cada vértice u adjacente a v, se $distances[v] + 1 > distances[u]$ se sim então $distances[u] = distances[v] + 1$. No final o valor máximo do vetor *distances* é o nosso resultado final.

Análise Teórica

Pseudo Código com complexidade

- Leitura dos dados de entrada e criação do grafo G e G^T a depender linearmente de E(número de relações): $\mathcal{O}(E)$
- Processamento do grafo G através do algoritmo $DFS(G)+DFS(G^T)$ de forma a eliminar loops e criar os grupos SCC: $\mathcal{O}(V + E)$
- Criação do grafo G^{SCC}: $\mathcal{O}(V + E)$
- DFS sobre o grafo G^{SCC} para obter ordem topológica, sendo n o número de grupos SCC e m o número de arcos do grafo G^{SCC}: $\mathcal{O}(n + m)$
- Algoritmo de procura de caminho mais longo: $\mathcal{O}(n + m)$
- Apresentação dos dados $\mathcal{O}(1)$

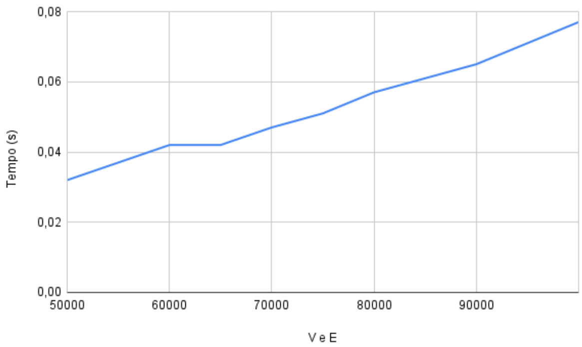
Complexidade global da solução: $\mathcal{O}(V + E)$

Relatório 2º projeto ASA 2023/2024

Grupo: TP049
Aluno(s): Henrique Machado (103202) e Vasco Pereira (103368)

Avaliação Experimental dos Resultados

Como analisada antes a complexidade temporal do algoritmo apenas depende do número de indivíduos(V) e do número de relações entre indivíduos(E), por isso foram gerados testes 10 testes com V e E iguais a começar em 50000 e incrementando em intervalos de 5000, podemos observar o gráfico de tempo no Gráfico 1 e a tabela de resultados na Tabela 1.



(a) Gráfico 1

V, E	Real Time (s)
50000	0,032
55000	0,037
60000	0,042
65000	0,042
70000	0,047
75000	0,051
80000	0,057
85000	0,061
90000	0,065
95000	0,071
100000	0,077

(b) Tabela 1

Como conseguimos ver o gráfico gerado está de acordo com a complexidade teórica sendo que o tempo de execução é linear em relação a V e E.
Logo se pusermos o eixo dos X a variar com a complexidade prevista na analise teórica ($\mathcal{O}(V + E)$) o gráfico será também linear como pode ser visto no gráfico 2.

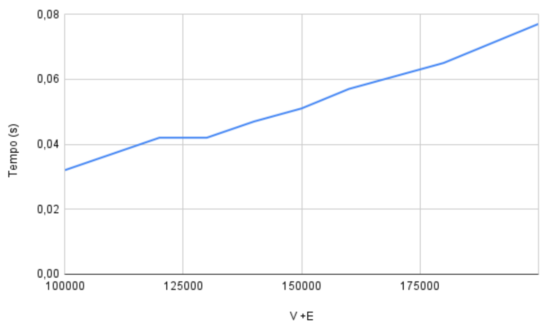


Figure 2: Gráfico 2

Assim conseguimos ver que temos uma relação linear com os tempos no eixo dos Y, confirmando que a nossa implementação está de acordo com a analise teórica.