**Descrição do Problema e da Solução (Necessita revisão gramatical)**

Problema: Assumindo que os alunos partilham resumos, formas de estudo e soluções com os seus amigos, o professor da disciplina em questão considerou que a nota dos alunos devia ter em conta as notas dos seus amigos. Então, foi-nos proposto que desenvolvêssemos um algoritmo otimizado que alterasse as notas previamente obtidas pela nota máxima das relações sociais de cada aluno.

Solução: Para representar o grafo e as suas conexões utilizámos listas de adjacências. A nossa solução baseia-se em utilizar o algoritmo de Tarjan para identificar as SCC do grafo e garantir que os filhos independentes (se existirem) são verificados antes dos seus predecessores. Criámos uma função que, no momento do Pop do algoritmo de Tarjan, atua de modo a alterar a nota máxima dos vértices das SCCs, se necessário.

A função verifica dois casos: Se o vértice for independente e o seu outdegree for diferente de 0, (como o algoritmo de Tarjan garante que as suas ligações já foram verificadas), trocamos a sua nota com a nota máxima dos filhos, se necessário; No caso de estarmos na presença de um vértice independente com outdegree igual 0 (o aluno não tem relações sociais), não acontece nada, porque não é possível a sua nota vir a ser alterada; No caso de ser um SCC, obtém o máximo das notas dos vértices pertencentes à SCC (juntamente com os seus vértices-filho) e troca para a maior obtida.

**Análise Teórica**

1. Leitura dos dados de entrada: nesta função temos dois for’s: um para colocar no grafo os vértices (armazenando o espaço necessário), e outro para fazer as conexões entre os vértices. Assim, temos um for com complexidade Θ(V) e outro com complexidade Θ(E), pelo que no final a complexidade da leitura é Θ(V+E).
2. Processamento do grafo para fazer alguma coisa. Logo, O(??)
3. Aplicação do algoritmo de Tarjan. Como dito no resumo da solução, utlizamos esse algoritmo para identificar as SCCs do gráfico e garantir que os filhos independentes, se existirem, são verificados antes dos seus predecessores.

Chamadas à função tarjan recursivamente → O(V)

Lista de adjacências de cada vértice é verificada apenas uma vez → O(E)

Logo, a aplicação do Tarjan → O(V+E)

Análise teórica da complexidade total e das várias etapas da solução proposta

Inserir aqui um pseudo código de muito alto nível a indicar a complexidade de cada etapa.

Exemplo:

* Leitura dos dados de entrada: simples leitura do input, com ciclo(s) a depender de linearmente/quadraticamente/… de V/E/V+E/… Logo, Θ(V)
* Processamento do grafo para fazer alguma coisa. Logo, O(??)
* Aplicação do algoritmo X para fazer algo. Logo, O(?X?X)
* Transformação dos dados com uma dada finalidade. O(?Y?Y?)
* Apresentação dos dados. O(???)

Complexidade global da solução: O(!??!)

**Avaliação Experimental dos Resultados**

Descrição do tipo experiências feitas e gráfico demonstrativo da avaliação de tempos associados.

Gerar vários grafos de tamanho incremental e cálculo dos tempos para cada instância. Gerar o gráfico do tempo em função do tamanho do grafo de entrada como exemplificado abaixo.



Concluir se o gráfico gerado está concordante com a análise teórica prevista.