

Relatório 1º projecto ASA 2019/2020

Grupo: al112

Aluno(s): Duarte Bento (92456) e Susana Monteiro (92560)

Descrição do Problema e da Solução

Foi dado um conjunto de alunos e respetivas notas finais, estimadas apenas a partir da sua classificação no primeiro teste. Além disso, foi dado um conjunto de relações de amizade (não recíprocas) tal que um aluno A que tenha como amigo um aluno B ficará com a maior nota entre os dois. No entanto, isto não significa que o contrário se aplique, visto que as relações são direcionadas. Assim, foi pedido que, tendo em conta as notas iniciais e relações de amizade entre alunos, fosse desenvolvido um programa que determinasse a nota final de cada aluno.

Deste modo, optou-se solucionar o problema recorrendo a um grafo dirigido, em que os vértices correspondem aos alunos e as arestas às relações entre estes (se A tem como amigo B, o vértice A tem uma aresta para o vértice B). Cada vértice guarda o valor da nota do aluno, valor que vai sendo atualizado ao longo da execução. A solução proposta passa por fazer uma procura em profundidade, recorrendo a recursão, percorrendo apenas uma vez cada aresta. Quando se atinge um vértice que não tem mais amigos por visitar, termina-se esse vértice e considera-se que a nota atual do aluno é a final. Porém, no caso de existirem ciclos no grafo, observou-se que a nota que o aluno tem no momento em que o seu vértice é terminado pode não corresponder à nota máxima que ele consegue atingir. Para corrigir este erro, decidiu-se repetir o processo até que não sejam verificadas alterações no grafo, garantido assim que todos os alunos têm a nota máxima que poderiam ter.

A visita de cada vértice consiste em, primeiramente, verificar se o vértice já foi ou não visitado. No primeiro caso, retorna-se logo o valor da sua nota (sendo ele o maior valor que o vértice pode ter, excetuando em casos com ciclos). No segundo, marca-se o vértice como visitado, visitam-se todos os vértices adjacentes a este e compara-se o valor do próprio com o resultado obtido das visitas, atribuindo o maior destes valores como o novo valor do vértice visitado.

Análise Teórica

- Criação do grafo: $\Theta(V+E)$
 - Leitura das notas (vértices) e inserção na cabeça da lista de adjacências, ciclo que depende linearmente de V, logo $\Theta(V)$;
 - Leitura das relações (arestas) e introdução na lista de adjacências, ciclo que depende linearmente de E, logo $\Theta(E)$;
- Aplicar algoritmo enquanto forem registadas alterações (uma vez no melhor caso - quando todos os alunos já têm a melhor nota que poderiam ter - e estima-se que, no pior caso, esse valor seja igual ao número de vértices), portanto $O(V) * [O(V+E) + O(V)] = O(V(V+E))$
 - Visita dos vértices percorrendo todas as arestas e todos os vértices atualizando o valor, logo $O(V+E)$
 - Repor o estado dos vértices (não visitado), logo $O(V)$
- Apresentação do resultado final (valor de cada vértice). $\Theta(V)$

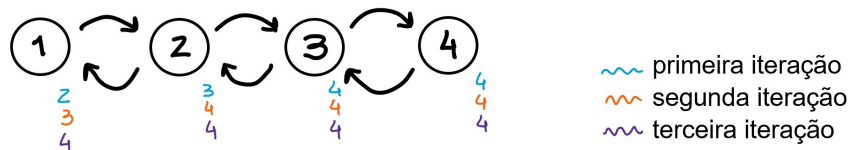
Complexidade global da solução: $O(V+E) + O(V(V+E)) + O(V) = O(V(V+E))$

Relatório 1º projecto ASA 2019/2020

Grupo: al112

Aluno(s): Duarte Bento (92456) e Susana Monteiro (92560)

Para suportar a alegação de pior caso, resolveu incluir-se um exemplo que escala até $O(V(V+E))$. Começando no vértice 4:

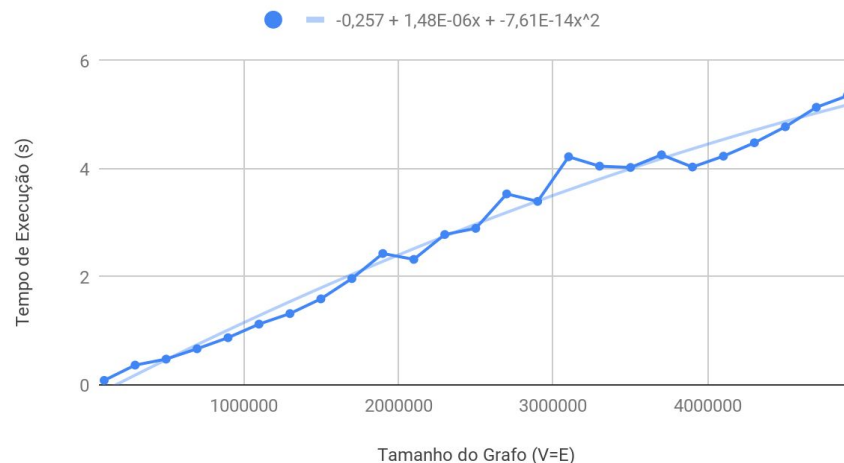


Aplicando o algoritmo desenvolvido neste exemplo, com quatro vértices ($\#V$), foi necessário quatro iterações ($\#V$). É de notar que este é o pior caso do algoritmo e que no geral, como irá mais a frente ser analisado (resultados experimentais), o número de iterações é reduzido levando a uma complexidade praticamente linear.

Avaliação Experimental dos Resultados

Recorrendo a um script, começou-se por gerar inputs aleatórios de modo a reunir um conjunto de valores que pudesse ser estudado. No entanto, verificou-se que fazendo variar não só o número de vértices e de arestas, mas também o número de sub-redes e o número máximo de alunos por sub-rede, obteve-se uma grande discrepância entre valores, o que dificultava o estudo do gráfico. Deste modo, decidiu-se fazer apenas variar o número de vértices e de arestas para que melhor pudesse ser estudado o efeito destes números (que afetam diretamente a complexidade) no tempo de execução do algoritmo nos grafos gerados.

Variação do Tempo de Execução em ordem ao Tamanho do Grafo



Concluimos que, embora a complexidade deste algoritmo no pior caso seja quadrática, em geral esta é praticamente linear. Aliás, como se pode observar no gráfico, a função de tendência é quase linear.