

# Relatório 1º projeto ASA 2019/2020

**Grupo:** tp020

**Aluno:** João Miguel Pipa Ferreira Caldeira (93729)

---

## Descrição do Problema e da Solução

O problema apresentado consiste no aprimoramento de um algoritmo feito para prever as notas do projeto dos alunos da disciplina ASA tendo por base a nota que obtiveram no primeiro teste. Deseja-se que o aprimoramento no algoritmo considere a partilha de soluções que existe entre amigos na turma para estimar uma nota de um aluno X.

A solução proposta consiste em criar um grafo com os alunos e as suas relações de amizade e em seguida realizar uma DFS em cada ponto desse grafo de forma a percorrer todos os possíveis caminhos, partindo desse ponto e comparando sempre as notas na relação a percorrer e atualizando a nota mais baixa para a nota mais alta. Esta solução tem por código base, um código de uma DFS iterativa, que chama a DFS para cada vértice tendo sido adaptado para o problema em causa. O código base encontra-se neste [link](#).

## Análise Teórica

- A leitura dos dados de entrada está dividida em 3 partes, primeiro obtém-se o número de alunos e o número de relações da primeira linha. Em seguida e aliado à construção do grafo, obtém-se as notas dos alunos através da função “trataAlunos” com um ciclo a depender linearmente de V (número de alunos),  $\Theta(V)$ . Por fim, e completando também o grafo, obtém-se as relações entre os vértices através da função “trataRelacoes” com um ciclo a depender linearmente de E (número de vértices),  $\Theta(E)$ . Concluimos assim que a obtenção dos dados de entrada tem uma complexidade, no seu total, de  $\Theta(V+E)$ .  
*obter número de alunos e relações*

*while número de alunos > contador:*  
    *obter nota*  
    *introduzir nota no grafo*

*while número de relações > contador:*  
    *obter relação*  
    *introduzir relação no grafo*

- Em seguida aplica-se a DFS em cada vértice do grafo com um ciclo a depender linearmente do número de vértices,  $\Theta(V)$  e para cada DFS aplicada num vértice tem-se um ciclo que depende linearmente das relações do vértice + número de vértices,  $\Theta(V+E)$ , que irá também transformar as notas dos alunos conforme o enunciado. Isto perfaz uma complexidade de  $\Theta(V*(V+E))$  na aplicação do algoritmo e tratamento de dados.  
*For contador < Nº de vértices*

# Relatório 1º projeto ASA 2019/2020

**Grupo:** tp020

**Aluno:** João Miguel Pipa Ferreira Caldeira (93729)

---

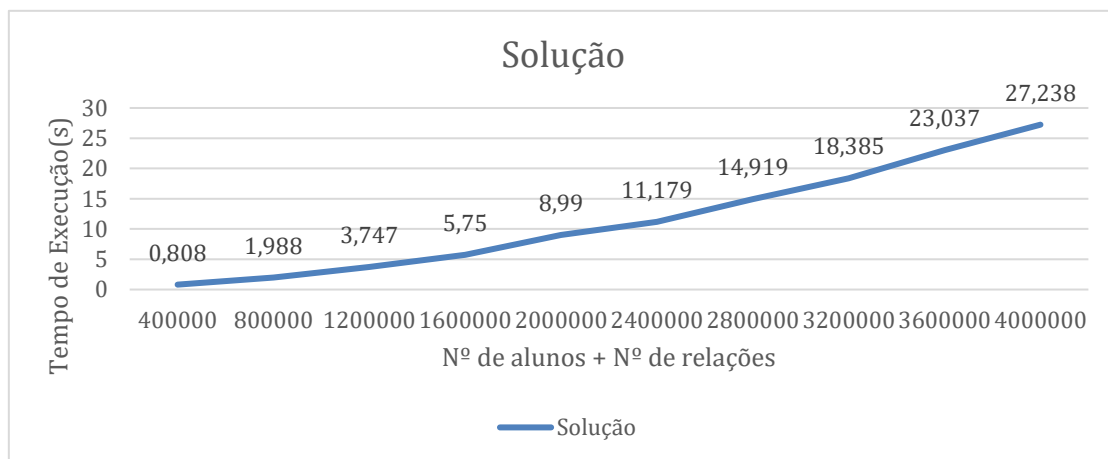
```
    If not visited:
        DFS(vértice, vetor)
DFS(v, vetor):
    While stack not empty:
        For iterar lista de adjacência:
            If nota de adjacente > nota de v:
                Reset da DFS no vértice v com atualização
                da nota
            If not visited and nota de adjacente inferior:
                colocar no stack para visitar
```

- A apresentação final dos dados no stdout é feita através de um ciclo que depende linearmente do número de alunos, ou seja,  $\Theta(V)$ .  
For iterar vetor de alunos:  
Imprimir a nota do aluno  $\ln$
- Logo a complexidade global da solução será  $\Theta(V*(V+E))$ .

## Avaliação Experimental dos Resultados

Foram gerados 10 grafos com número de vértices a variar de 100.000 a 1.000.000, e número de relações a variar de 300.000 a 3.000.000, mantendo esta relação para cada instância. O valor das sub-redes é para cada grafo (número de vértices) /5.

Depois de gerados esses 10 grafos, foi testado o algoritmo para cada um dos casos e com os tempos de execução foi construído o seguinte gráfico:



O gráfico gerado respeita a estimativa de complexidade  $\Theta(V*(V+E))$  já que se pode verificar uma curva em função do número de vértices e de relações dos grafos, notando-se um crescente declive nessa curva.