PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Ciência da Computação

Thiago Henrique Balbino Dias

**DOCUMENTAÇÃO**

**Trabalho Prático de Algoritmos em Grafos**

Contagem

2015

.A COMPLEXIDADE do algoritmo é O(n2)

.PROBLEMA:

* .Existe um arquivo contendo o nome de 20 áreas de pesquisa
* .Existe um arquivo contendo uma matriz com o grau de dissimilaridade dessas pesquisas
* .Existe um arquivo com informações do aluno em cada linha, ou seja, seu nome e sua área de pesquisa
* .Existe um número n de alunos
* .Existe um número k de professores (número de clusters/grupos)
* .Cada aluno está associado a uma única pesquisa
* .Mais de um aluno pode estar associado à mesma área de pesquisa
* .Se o aluno vi se relaciona com o aluno vj, então o aluno vj se relaciona com o aluno vi
* .Se dois alunos atuam em uma mesma área de pesquisa, o grau de dissimilaridade entre eles é zero

.SOLUÇÃO:

* .Foi utilizado uma matriz de adjacência, pois cada aluno se relaciona com todos os outros.
* .Os vértices são os alunos.
* .As arestas são o grau de dissimilaridade entre os temas de pesquisa.

.DESCRIÇÃO DO ALGORITMO:

* .O algoritmo utiliza o conceito de orientação a objeto.
* .O algoritmo possui um método chamado controle que é publico e chama os outros.
* .O controle começa chamando o método "lerNomePesquisa()" que lê o arquivo que contém os nomes das áreas de pesquisa, e armazena em um array do tipo "Pesquisa".
* .Depois chama o método "lerAluno()", que ler o arquivo "entrada.txt" e armazena os alunos em um array do tipo "Aluno".
* .Depois chama o método "montarMatrizDissimilar()", que ler o arquivo contendo esses dados e monta uma matriz.
* .Depois chama o método "montarGrafo()", que monta o grafo do problema, vértices = alunos, arestas = grau\_dissimilar.
* .Depois chama o método "montarArrayAresta()", que é um array que contém todas as arestas do grafo. Esse método chama outro método que ordena o arrayAresta.
* .Depois chama o método "montarAgm()", que monta a árvore geradora mínima utilizando o algoritmo de kruskal adaptado ao algoritmo do trabalho.
* .Depois chama o método "montarCluster()", que faz o devido corte na AGM e dá um "set" em cada objeto Aluno no campo "Aluno.setCluster(int cluster)".
* .Depois chama o método "mostrarCluster()", que apresenta os resultados encontrados.
* .O trabalho utilizou o algorimo de busca em profundidade por causa do algoritmo de kruskal, para evitar circuito na AGM.
* .O trabalho utilizou o algoritmo busca em largura para montar os clusters.
* .O método de ordenação utilizado foi o de inserção.

**.OBS:**

* .Para se executar o programa, dentro da pasta deve conter um arquivo chamado "entrada.txt", que deve conter 1..10..30..50.. ou mais dados de entrada.
* Este trabalho está disponível em:
  + - <https://github.com/thbdias/tp_grafo>