

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Engenharia de Computação - Algoritmos em Grafos

Gustavo Loschi Salomão

Ryan Lemes Bezerra

PROJETO FINAL

GRAFOS

Belo Horizonte

2017

Gustavo Loschi Salomão

Ryan Lemes Bezerra

PROJETO FINAL

GRAFOS

Projeto final apresentado à matéria de algoritmos em grafos do programa de graduação em engenharia de computação da Pontifícia Universidade Católica de Minas Gerais.

Orientadora: Eveline Alonso Veloso

Belo Horizonte

2017

SUMÁRIO

1 INTRODUÇÃO	3
2 MODELAGEM E SOLUÇÃO	3
2.1 MODELAGEM	3
2.2 SOLUÇÃO	3
3 TESTES	4
3.1 DADOS ALUNO TEMA	4
3.2 NOME TEMA PESQUISA	4
3.3 MATRIZ DE DISSIMILARIDADE	4
3.4 VALORES ESPERADOS	5
3.5 TESTE NO PROGRAMA	6
4 CONCLUSÃO	7

1 INTRODUÇÃO

Este trabalho tem como objetivo modelar e resolver um programa para encontrar o conjunto de grupos de orientação mais adequado na matéria de trabalhos de diplomação II, aplicando teoremas em grafos. Onde em um mesmo grupo, os alunos integrantes terão temas mais semelhantes.

2 MODELAGEM E SOLUÇÃO

O problema foi modelado em grafos e depois foi implementado na linguagem C#.

2.1 MODELAGEM

O problema foi modelado, onde os vértices são os temas, e as arestas só ligam os temas que são dissimilares. Desse modo todos os temas serão ligados entre si, uma vez que todos os temas possuem alguma dissimilaridade, mas não haverá loops, já que um tema não é dissimilar a ele mesmo. Os pesos das arestas são as dissimilaridades entre os temas.

2.2 SOLUÇÃO

Para separar os grupos mais adequadamente, foi criado, inicialmente, um grafo completo. Dessa forma é possível observar qual o nível de dissimilaridade entre todos os temas.

A partir desse grafo completo, foi gerada a árvore geradora mínima, utilizando o algoritmo de Kruskal. Consequentemente, o grafo obtido não terá ligação entre temas muito dissimilares.

Por fim, para separar os grupos, retira-se da árvore geradora mínima $(K - 1)$ arestas de maior peso, onde K é o número de professores orientadores. Desse

modo o grafo será separado em K componentes conexos. Onde cada componente conexo será um grupo.

3 TESTES

Os testes foram realizados com os seguintes dados:

3.1 DADOS ALUNO TEMA

```
01 02
02 04
03 06
04 08
05 10
```

3.2 NOME TEMA PESQUISA

```
02 ALGORITMOS E ESTRUTURAS DE DADOS
04 REDE DE COMPUTADORES
06 INTRODUCAO A PESQUISA EM INFORMATICA
08 ARQUITETURA E ORGANIZACAO DE COMPUTADORES
10 ALGORITMOS EM GRAFOS|
```

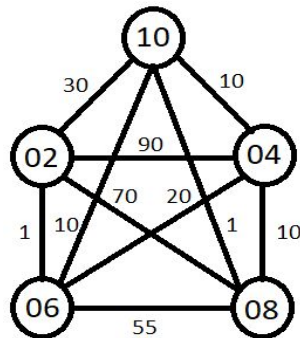
3.3 MATRIZ DE DISSIMILARIDADE

```
0 90 1 70 30
90 0 20 10 10
1 20 0 55 10
70 10 55 0 1
30 10 10 1 0
```

3.4 VALORES ESPERADOS

Modelando os arquivos em grafos iremos obter o seguinte grafo:

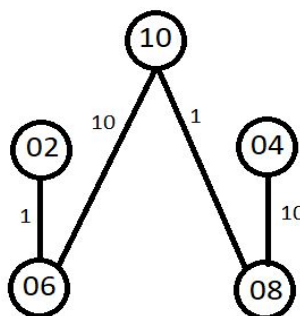
Figura 1 - Grafo Completo



Fonte: criada pelos autores

Aplicando o algoritmo de Kruskal iremos obter a seguinte árvore geradora mínima:

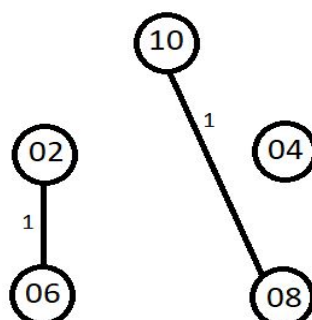
Figura 2 - Árvore Geradora Mínima



Fonte: criada pelos autores

Após encontrada a árvore geradora mínima serão retiradas $(K - 1)$ arestas de maior peso. Que resultará em um grafo com três componentes conexos, já que no nosso teste teremos três professores orientadores. Resultando no seguinte grafo:

Figura 3 - Componentes Conexos



Fonte: criada pelos autores

Os componentes conexos do grafo serão os grupos. Dessa forma teremos três grupos:

- Grupo 1 => Aluno: 01
 Tema: Algoritmo e Estrutura de Dados
 Aluno: 03
 Tema: Introdução à Pesquisa em Informática
- Grupo 2 => Aluno: 04
 Tema: Arquitetura e Organização de Computadores
 Aluno: 05
 Tema: Algoritmos em Grafos
- Grupo 3 => Aluno: 02
 Tema: Rede de Computadores

3.5 TESTE NO PROGRAMA

Ao rodar no programa com os mesmos arquivos, obtém-se o seguinte resultado:

Figura 4 - Teste no Programa

```
Trabalho Final Grafos
Integrantes:
~> Gustavo Loschi Salomao
~> Ryan Lemes Bezerra
-----
Digite a quantidade de professores orientadores: 3

Grupo 1:
Aluno: 3
Tema: INTRODUCAO A PESQUISA EM INFORMATICA
-----
Aluno: 1
Tema: ALGORITMOS E ESTRUTURAS DE DADOS
-----

Grupo 2:
Aluno: 2
Tema: REDE DE COMPUTADORES
-----

Grupo 3:
Aluno: 5
Tema: ALGORITMOS EM GRAFOS
-----
Aluno: 4
Tema: ARQUITETURA E ORGANIZACAO DE COMPUTADORES
-----
```

fonte : criada pelos autores

É possível observar que o resultado obtido no programa é o mesmo do que o esperado. Portanto podemos afirmar que o programa funciona corretamente para o problema proposto.

4 CONCLUSÃO

Aplicando teoremas em grafos ao problema apresentado, foi possível moldar o problema e implementá-lo. A implementação feita na linguagem C#, funcionou conforme o esperado, já que os resultados obtidos com a execução do programa foi exatamente o resultado esperado.