



**UNIVERSIDADE ESTADUAL DE MARINGÁ – UEM  
CENTRO DE TECNOLOGIA  
DEPARTAMENTO DE INFORMÁTICA**

Disciplina: ALGORITMOS EM GRAFOS  
Professor: DR. Daniel Kikuti

**ALGORITMOS EM GRAFOS**

**SOLUÇÃO DO PROBLEMA:**  
**“ESTRADAS ESCURAS – URI 1152”**

ALUNO:

THIAGO RODRIGO BUCALÃO

Maringá, Setembro de 2015.



## 1. Descrição do problema:

O problema escolhido para se resolver foi o das “Estradas Escuras – URI 1152”, consiste em otimizar o gasto de energia através do desligamento de algumas rodovias que ligam as cidades, entretanto deve haver um caminho seguro que liguem as “n” cidades. Ao final deve se dizer qual foi o valor que pode ser economizado.

A representação do grafo será:

- Cada cidade é um vértice do grafo;
- Cada estrada é uma aresta;

Exemplo:

Entrada

6 9

0 5 2

0 1 5

0 4 15

1 5 4

1 4 22

1 3 9

5 3 6

4 2 1

2 3 12

0 0

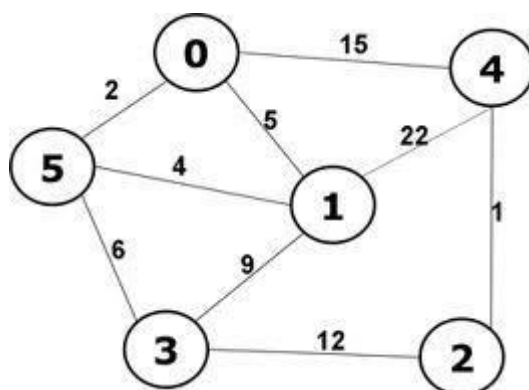


Figura: representação do grafo

## 2. Algoritmo utilizado:



Para se resolver este problema a técnica utilizada é encontrar uma Árvore Geradora Mínima (MST), ao qual é possível encontrar o menor caminho possível em um grafo saindo de um vértice  $k$  qualquer e acessar a qualquer outro vértice do grafo.

Pseudocódigo:

```
GENERIC-MST( $G, w$ )
1   $A = \emptyset$ 
2  while  $A$  does not form a spanning tree
3      find an edge  $(u, v)$  that is safe for  $A$ 
4       $A = A \cup \{(u, v)\}$ 
5  return  $A$ 
```

Portanto essa técnica possui a característica, pois escolhe a menor aresta no momento da iteração, ou seja, uma aresta segura e que não forme ciclo na árvore geradora. Isso garante que ao final teremos o menor caminho possível conforme o (Cormen p.448) nos prova.

### 3. Estrutura de Dados utilizada e análise de complexidade:

- Struct:  
Utilizado para criar a estrutura do grafo que é composta por vértice de saída, vértice de chegada e peso da aresta. Essa estrutura facilita, pois, agrupa as informações que um grafo contém.
- Malloc:  
Foi utilizado para alocar espaço na memória para se criar uma lista compostas dos vértices e arestas do grafo. Essa estrutura é fundamental pois, não é possível inicializar o valor máximo de vértices e arestas isso é feito em tempo de execução.
- Union:  
Foi feita para realizar a comparação de elementos no conjunto, caso não pertença ao conjunto então acrescentamos a nova aresta a solução. Para isso é feito uma comparação na lista que compõem os elementos do conjunto.
- Qsort:  
Utilizado para ordenar as arestas conforme o seu peso. Esse método de ordenação foi utilizado pois realiza a ordenação de forma bastante eficiente.

### 4. Casos de teste:

Foram feitos três casos de teste:

- I. Verificação se os dados de entrada satisfazem a condição do problema, caso não façam a execução deve ser interrompida;
- II. Verificação se o formato de entrada está sendo satisfeitos para continuar a execução do programa;
- III. Verificação para termino da execução ao qual deve ocorrer se, e somente se o programar receber  $\{0\}$  como entrada e o total de interações tenham sido satisfeitos.

