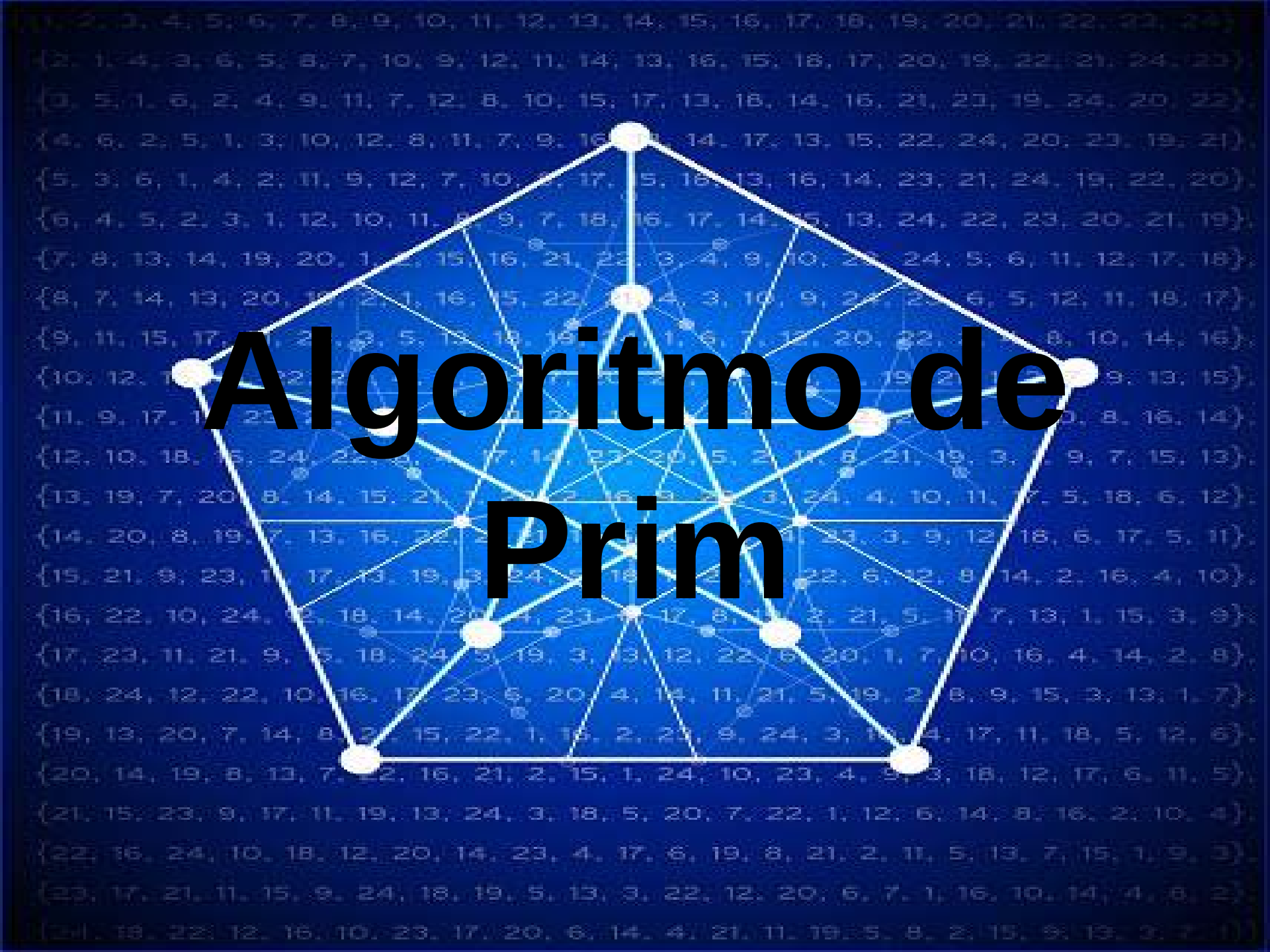
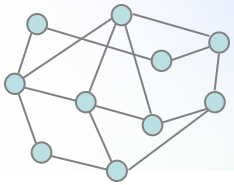

Algoritmo de Prim

Universidade Federal de Goiás
Campus de Catalão
Disciplina de Teoria dos Grafos

Algoritmo de Prim

Alunos:
Cicero Mauricio
Edson Gomes
Fernanda Neiva
Jander Diego
Raina Marques
Simone Alberto
Vinicios



Criação do algoritmo

- Em 1930 foi descoberto por um matemático: Vojtěch Jarník.
- Em 1957 por um cientista da computação: Robert Clay Prim .
- Em 1959 redescoberto pelo matemático: Edsger Dijkstra.

Robert Clay Prim

- Nasceu em 1921 na cidade de Sweetwater no Texas.

Formação Profissional

- Formou-se em matemática e ciência da computação.
- Em 1941 recebeu seu BS em Engenharia Elétrica da Universidade de Princeton .
- Em 1949 recebeu seu Ph.D. em Matemática da Universidade de Princeton.

Formação profissional

- De 1948 a 1949 trabalhou na Universidade de Princeton como um associado de pesquisa.

- Segunda Guerra Mundial(1941-1944) trabalhou como engenheiro.

- De 1944 até 1949 trabalhou para United States Naval como engenheiro.

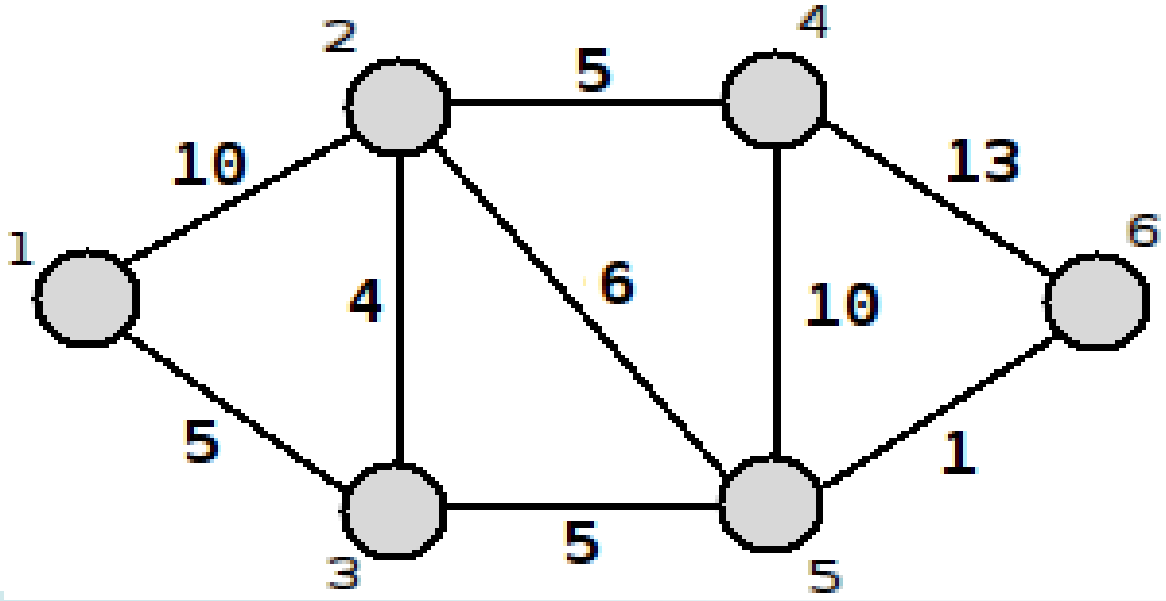
Formação profissional

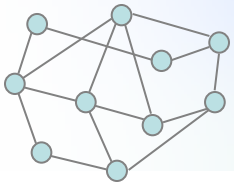
- De 1958-1961 atuou como diretor de pesquisa da matemática Bell Laboratories periodo que criou o algoritmo de Prim.

- Em 1962 tornou-se vice-presidente de pesquisa da Sandia National Laboratories.

Descrição do algoritmo

- O algoritmo de Prim ou DJP ou algoritmo de Prim-Jarník é uma aplicação em Teoria dos Grafos e sua função é encontrar a árvore geradora mínima para um grafo conexo com pesos.

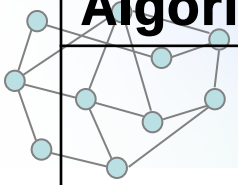




Descrição do Algoritmo

- Começa em um vértice até atar todos os vértices.
- Analisa a cada nó todas as possibilidades de transição e seus custos e escolhe o menor deles.
- Vai analisando a cada iteração o menor caminho e guarda-o na solução que será retornada no final.


Características:
Tabelinha da
salvação.



Algoritmo	Características
Prim	-Possui um ponto de partida; -Não pode ser orientado; -Os pesos podem ser iguais; -Combinação linear entre os vértices -Grafo Conexo; -Acíclico;
Dijkstra	-Possui um ponto de partida; -Pode ser orientado ou não; -O Custo das arestas não podem ser negativos; -Grafo Conexo
Kruskal	-Não possui um ponto de partida; -Não pode ser orientado; -Acíclico;
Boruvka	-Pesos distintos; -Alta velocidade de convergência; -Otimização combinatória; -Pode ser orientado ou não; -Não possui ponto de partida; -Busca a MST a partir da construção de uma floresta;

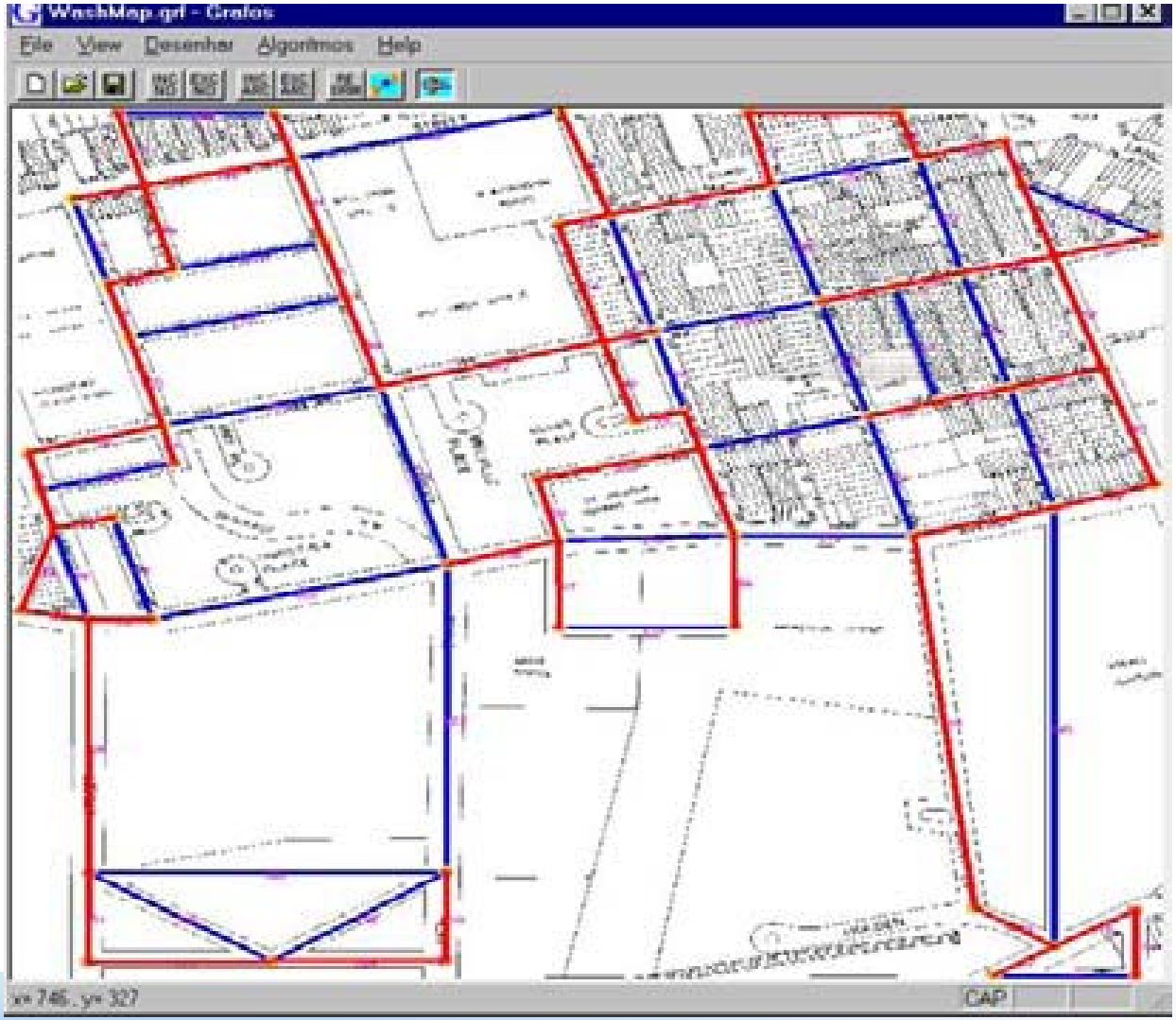
Aplicações:

- O algoritmo MENTOR para o projecto topológico de redes de telecomunicações Celso Lemos, Rui Valadas (REVISTA DO DETUA, VOL. 2, Nº 3, SETEMBRO 1998).

- Projeto do LAC/INPE voltado para a implementação de algoritmos em Sistemas Geográficos de Informação, como o software ARC-INFO (ESRI), em uso no projeto.

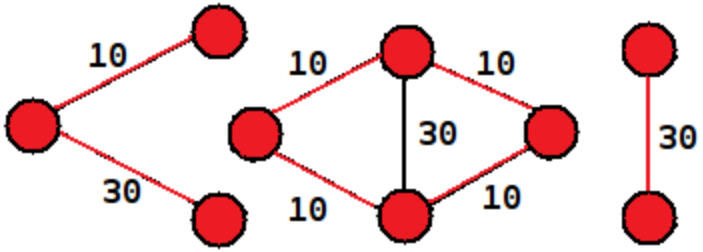
Algoritmo de prim em um mapa.

Imagem de satélite das rodovias feita pelo algoritmo prim

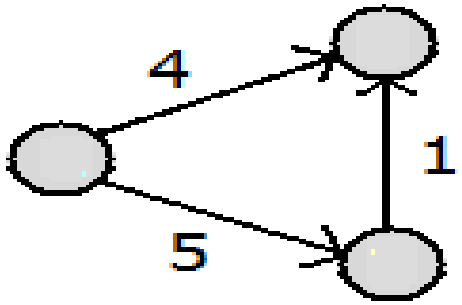


Casos em que o Prim não funciona:

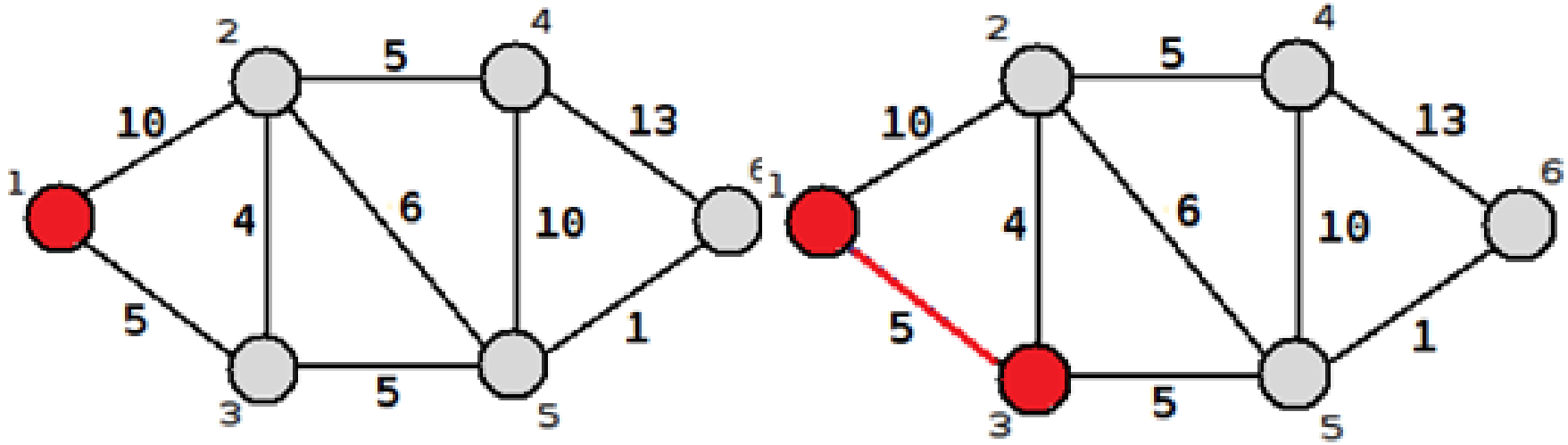
- grafos desconexos.



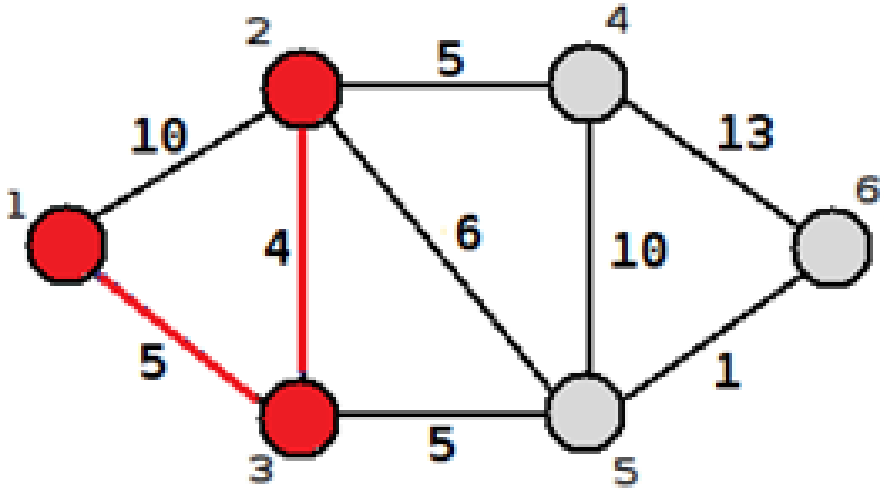
- grafos direcionais.

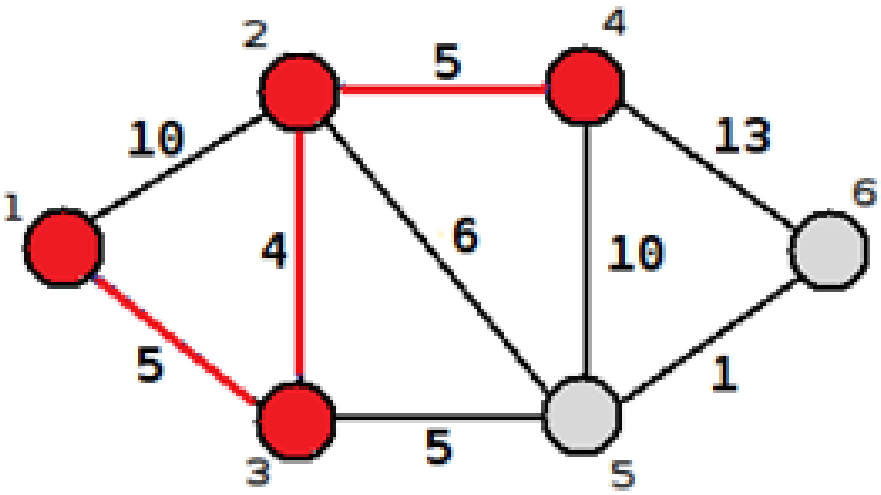


Como funciona?

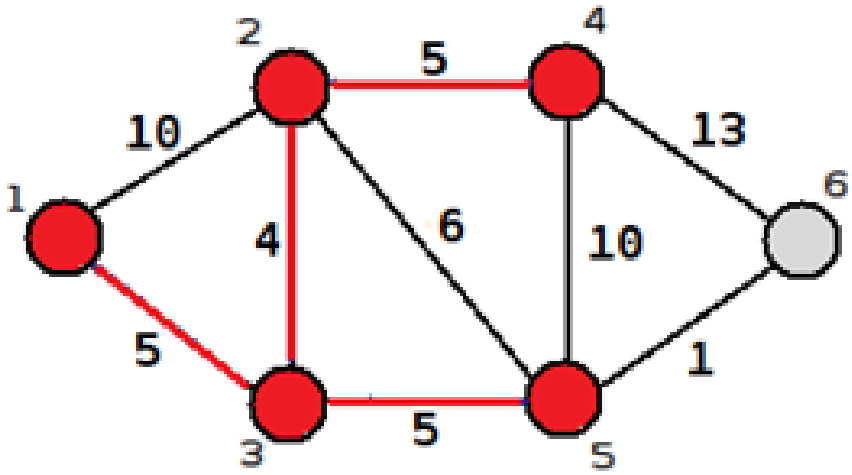


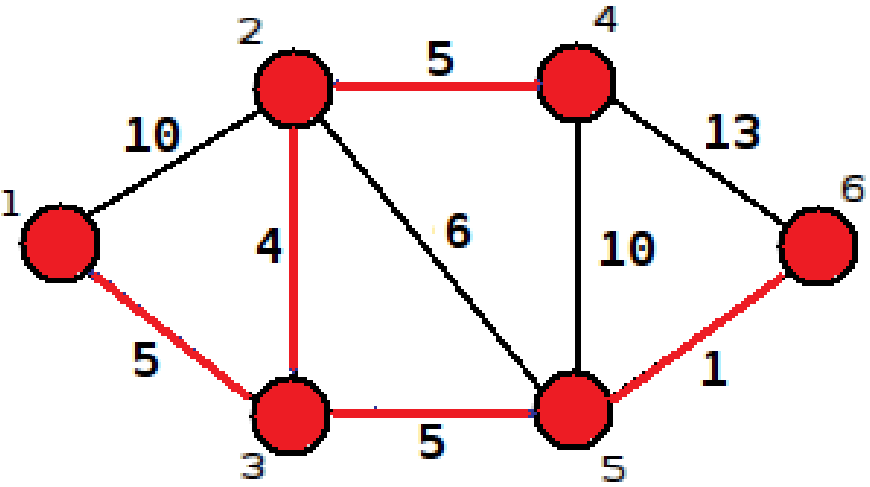
Como funciona?





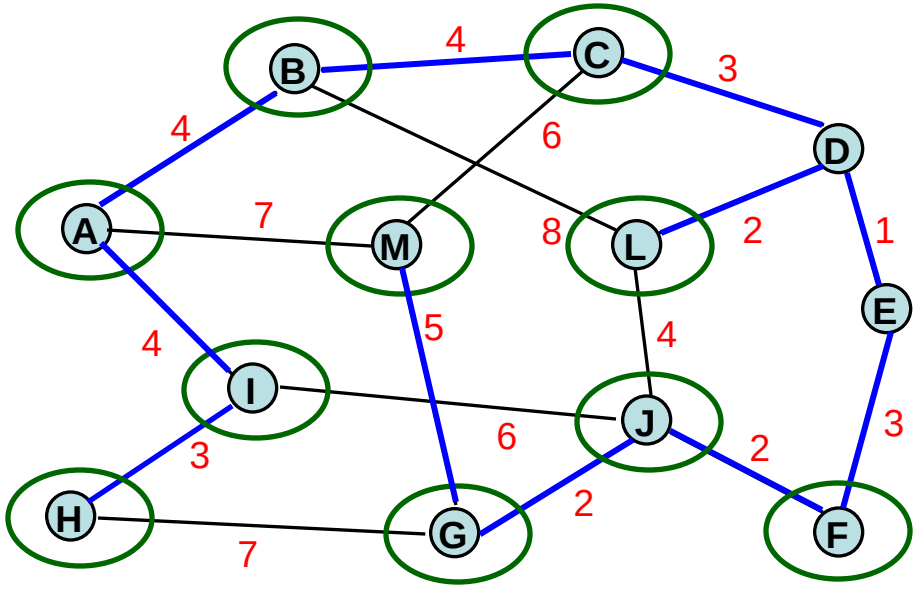
Como funciona?



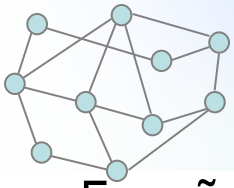


Árvore Geradora de Peso Mínimo

Exemplo:



c(F) = 33



Pseudocódigo

```
Função Prim(Grafo): inteiro
{
Declare:
        V[]:  inteiro
        E[]:  inteiro
        i:  inteiro
        k:  inteiro
        n_acabou: booleano
        menor:  inteiro
        custo: inteiro
        aux: inteiro
        total: inteiro
 i<-0
 k<-1
 V[0]<-0
```

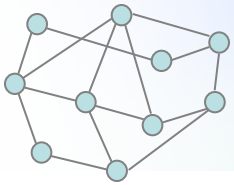
Pseudocódigo

```
montaMatrizAdjacência();
Enquanto(n_acabou) faça
{
    menor<- +∞

    aux <- 0
    Enquanto(aux < k) faça
{
        para j<-V[aux]+1 até tam-1
            {
             se((E[V[aux]][j] < menor) E (j não
                        pertencer a V) então
              {
                menor<-j
                custo<-A[V[aux]][j]
             }fim-se
```

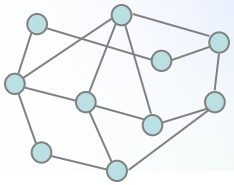
Pseudocódigo

```
                aux<-aux + 1
            }fim-para
            V[k]<-menor
            E[k-1]<-custo
            k<-k + 1
            se(k = tam) então
                n_acabou<-falso
            senão
                i<-menor
        }fim-enquanto
}fim-enquanto
para n<-0 até tam-2 faça
    total <- total + E[n]
 retorne total;
}FIM
```



Bibliografia

- Preiss, Bruno R. Estruturas de dados e algoritmos: Padrões de projetos orientados a objeto com java; tradução de Elizabeth Ferreira. Rio de Janeira: Campus, 2000. 519-521p.
- Goodrich, Michael T.; Tamassia, Robert. Estrutura de dados e algoritmos em java; tradução de Bernardo Copstein e João Batista Oliveira. 2 ed. Porto Alegre: Bookman, 2002. 549-550p.



OBRIGADO