INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Matemática Discreta

Leandro Colombi Resendo

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Algoritmos para Grafos

- Grafos Direcionados e Relações Binárias; o Algoritmo de Warshall
- Caminho de Euler e Circuito Hamiltoniano
- Caminho Mínimo e Árvore Geradora Mínima
- Algoritmos de Percurso

O Problema do Caminho Mínimo

Principais Algoritmos de Solução:

- Dijkstra – Obtém o caminho mínimo entre dois nós (não aceita arestas negativas).
- Ford-Bellman – Obtém o caminho mínimo entre um nó e todos os outro (Admite a existência de arestas negativas).
- Floyd-Warshall – Obtém o caminho mínimo entre todos os pares de nós (Admite a existência de arestas negativas).
- Yen – encontra os *K* caminhos mínimos entre todos os pares de nós.

Algoritmo de Dijkstra

1. Ler $G = (N,A)$, c_{ij} *(não negativo)* é a "distância" entre os nós e nó de origem s.
2. **Iniciar variáveis** $S := \emptyset$; $R := N$;
3. $d(i) := \infty\ \forall i \in N$; $d(s) := 0$ e $pred(s) := 0$;
4. **Enquanto** $|S| < n$ **fazer**
5. Seja $i \in N$ tal que $d(i) = min\{d(j): j \in R\}$;
6. $S := S \cup \{i\}$; $R := R - \{i\}$;
7. **Para** $j \in N$ com $(i,j) \in A$ **fazer**
8. **se** $d(j) > d(i)+c_{ij}$ **então** $d(j) = d(i)+c_{ij}$;
9. $pred(j) := i$;
10. Fim_Para
11. Fim_Enquanto

Algoritmo de Dijkstra

Por que funciona?

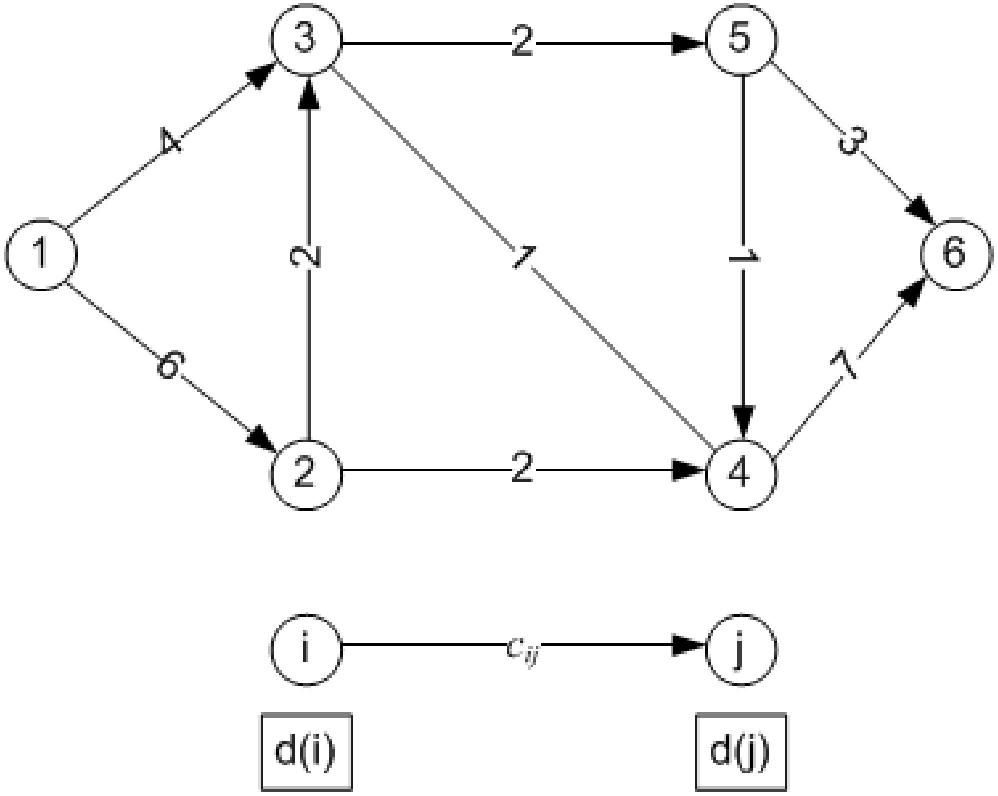
Note o seguinte trecho do código.

4. Enquanto $|S| < n$ fazer
5. Seja $i \in R$ tal que $d(i) = min\{d(j): j \in R\}$;
6. $S:=S \cup \{i\}$; $R:=R - \{i\}$;
7. Para $j \in N$ com $(i,j) \in A$ fazer
8. Se $d(j) > d(i)+c_{ij}$ então $d(j) = d(i)+c_{ij}$;
9. $pred(j):=i$;
10. Fim_Para
11. Fim_Enquanto

Resposta: A partir de um nó o algoritmo percorre, através dos caminhos mínimos, todas da redes buscando o nó de destino.

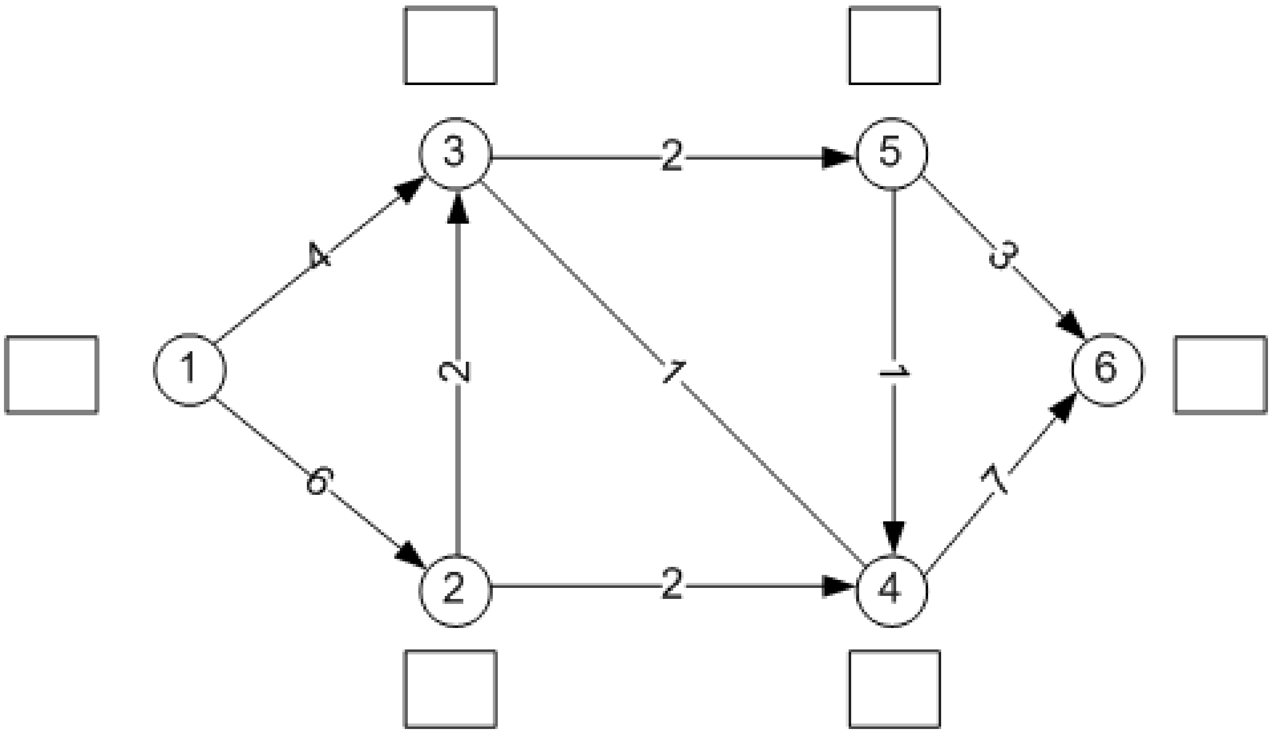
Algoritmo de Dijkstra

Exemplo:



Algoritmo de Dijkstra

Exemplo:



Árvores

Problema clássico: **Árvore geradora minima**.

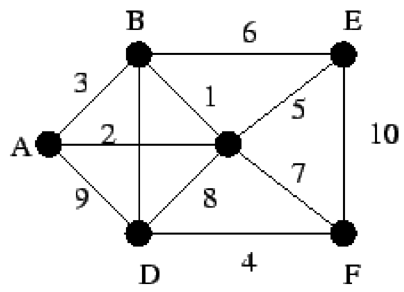
Considere uma rede não-direcionada (grafo) e associado a cada arco um custo (distância, tempo, etc) não-negativo. O objetivo é encontrar o caminho mais curto de tal maneira que os arcos formem um caminho entre todos os pares de nós

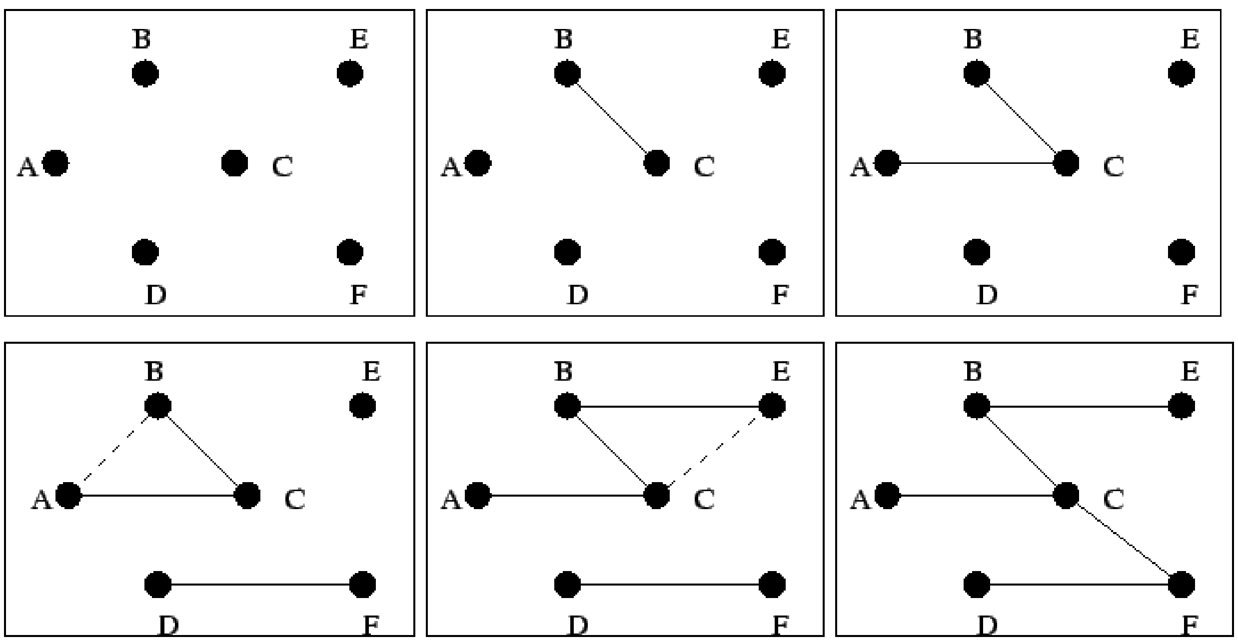
Exemplos de aplicações:

- Projeto de redes de telecomunicações (redes de computadores, redes de fibra-óptica, telefonia, televisão a cabo, etc).
- Projeto de rodovias, ferrovias, etc.
- Redes de transmissão de energia.

Árvores

Exemplo:





Árvores Geradora Mínima

<u>Principais Algoritmos de Solução:</u>

- Prim : monta a árvore mínima até inserir todos os nós.
- Complexidade $O(n^3)$.

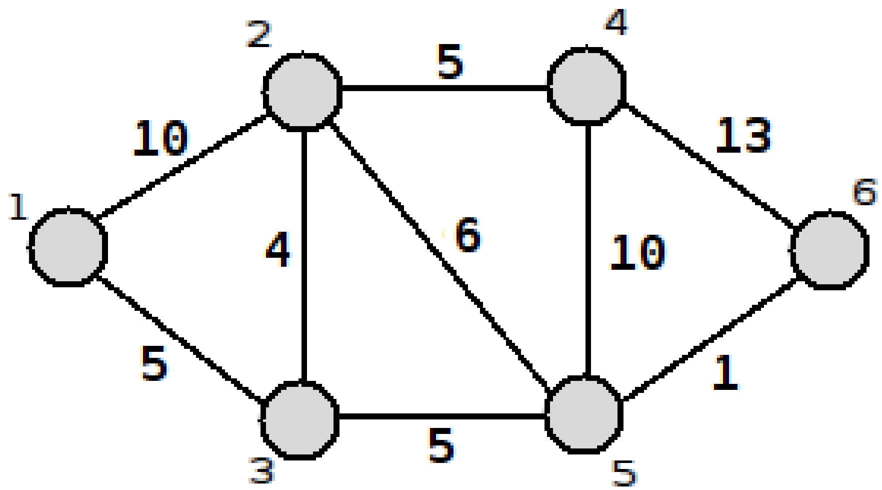
- Kruskal : seleciona as arestas em ordem crescente até obter uma árvore com todos os nós.
- Complexidade $O(m \log m)$.

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cariacica

Prim

Exemplo:

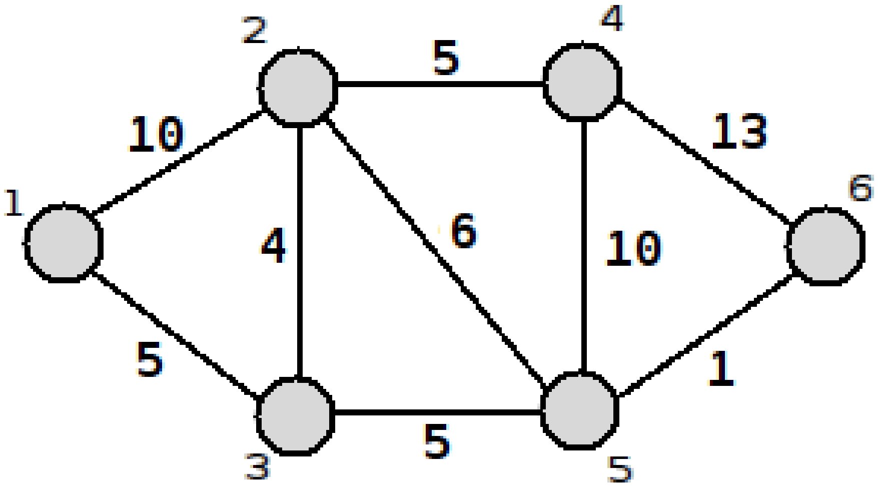
1. **Ler** $G = (N,A)$, d_{ij} é a “distância” entre os nós vizinhos.
2. Escolha qualquer vértice i.
3. **Iniciar variáveis** $T \leftarrow i; V \leftarrow N - \{i\};$
4. **Enquanto** $T \neq N$ **Para todo** $i \in T$ **fazer**
5. **Encontrar a menor aresta** $(i,k) \in A$ com $i \in T$ e $k \in V$.
6. $T \leftarrow T + \{k\}$
7. $V \leftarrow V - \{k\}$
8. $S \leftarrow S + \{(i,k)\}$
9. **Fim_Enquanto**
10. **Escrever** S (arestas da árvore geradora mínima)



Kruskal

Exemplo:

1. **Ler** $G = (N,A)$, d_{ij} é a “distância” entre os nós vizinhos.
2. Ordene os enlaces em ordem não-decrescente de distância (d_{ij}) no vertor $H = (h_i)$.
3. **Iniciar variáveis** $T \leftarrow h_1$; $i \leftarrow 2$;
4. **Enquanto** $|T| < n$ **Tome** $h_i \in H$ **fazer**
5. Se $T + h_i$ é um grafo acíclico (árvore) então
6. $T \leftarrow T + h_i$
7. $i \leftarrow i+1$
8. *Caso Contrário*
9. $i \leftarrow i+1$
10. **Fim_Enquanto**
11. **Escrever** T (arestas da árvore geradora mínima)



Complexidade

Prim:

4. **Enquanto $T \neq N$ Para todo $i \in T$ fazer**
5. **Encontrar** a menor aresta $(i,k) \in A$ com $i \in T$ e $k \in V$.
6. $T \leftarrow T + \{k\}$

Como "**Encontrar** a menor aresta" tem tempo *O(m)* e m é $O(n^2)$, repetindo *N* vezes do "Enquanto", temos um complexidade $O(n^3)$.

Kruskal

2. **Ordene** os enlaces em ordem não-crescente de distância (d_{ij}) no vertor $H = (h_i)$.
3. Iniciar variáveis $T \leftarrow h_1$; $i \leftarrow 2$;
4. **Enquanto** $|T| < n$ Tome $h_i \in H$ fazer

A "ordenação" das arestas pode ser feita em *O(mlogm)*, adicionando o *O(m)* no enquanto a complexidade se mantem *O(mlogm)*.

Lista Mínima de Exercícios

Seção 6.3: 1, 5, 7, 9, 10, 11, 14, 15, 21, 27.