

INSTITUTO FEDERAL
Goiás

Instituto Federal de Goiás
Câmpus Goiânia

Bacharelado em Sistemas de Informação
Disciplina: Estrutura de Dados II

Grafos Ponderados

Prof. Ms. Renan Rodrigues de Oliveira
Goiânia - GO

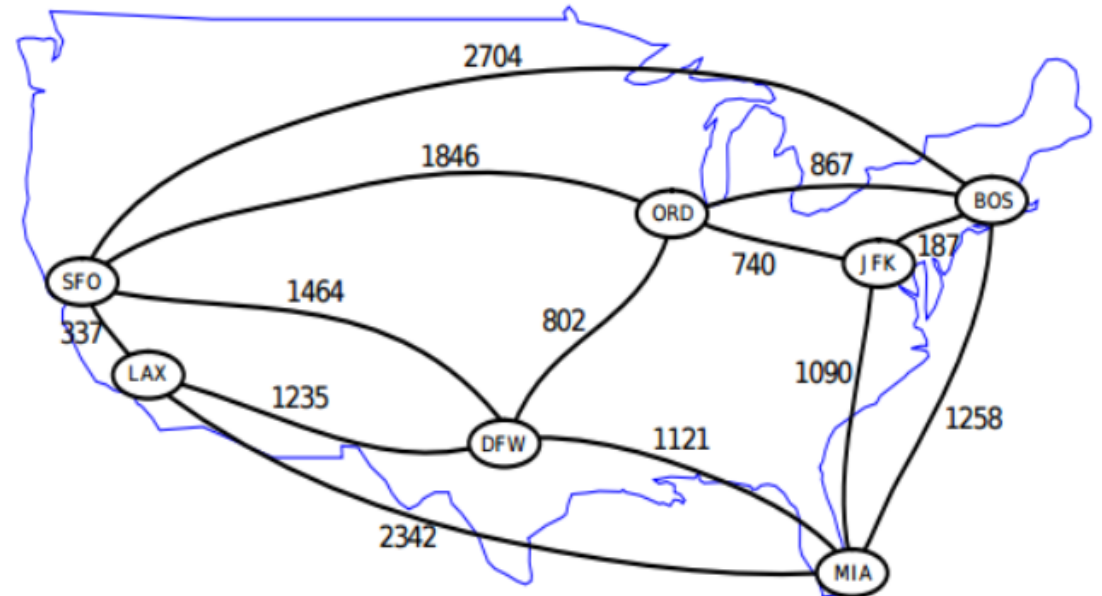
Introdução

Um grafo é ponderado quando suas arestas possuem um peso.



O significado do peso das arestas depende do problema.

- Por exemplo, se os nós em um grafo ponderado representa cidades, o peso das arestas poderia representar distâncias entre as cidades ou custos para voar entre elas.



Árvore Geradora Mínima para Grafos Ponderados



Para introduzir grafos ponderados, voltamos à questão da árvore geradora mínima.

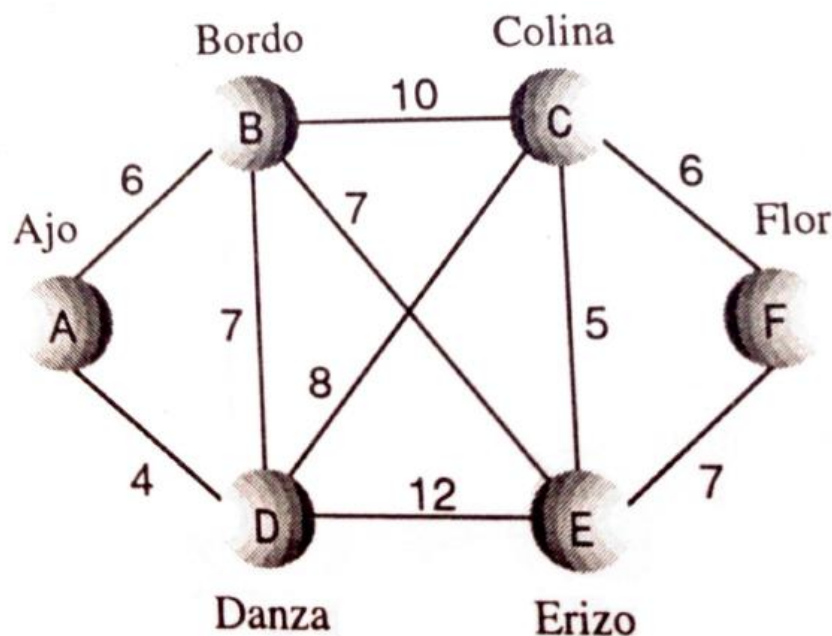
- ▶ Criar tal árvore é um pouco mais complicado com um grafo ponderado do que com um não ponderado.
- ▶ Quando todas as arestas têm o mesmo peso, é bem simples.
- ▶ Mas quando arestas podem ter pesos diferentes, alguma aritmética é necessária para escolher a correta.

Árvore Geradora Mínima



Suponha que queiramos instalar uma linha de televisão a cabo que conecte seis cidades.

- ▶ Cinco ligações conectarão as seis cidades, mas quais deveriam ser essas cinco ligações?
- ▶ No grafo abaixo, cada aresta tem um peso que representa o custo de milhões de dólares para instalação de uma ligação a cabo entre duas cidades.



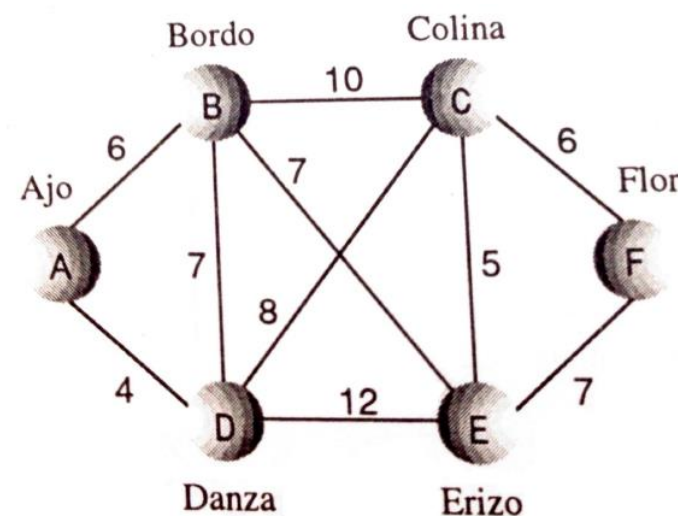
	A	B	C	D	E	F
A	inf	6	4	inf	inf	inf
B	6	inf	inf	10	7	inf
C	4	inf	inf	8	12	inf
D	inf	10	8	inf	inf	6
E	inf	7	12	inf	inf	7
F	inf	inf	inf	6	7	inf

Árvore Geradora Mínima



A maioria dos algoritmos de computador não “conhece” todos os dados de um certo problema de uma vez.

- ▶ Estes algoritmos tem que adquirir dados aos poucos, modificando a sua visão das coisas à medida que avança.
- ▶ Com grafos, algoritmos tendem a começar em algum nó e trabalhar para fora, adquirindo dados de nós próximos antes de descobrir nós mais distantes.
- ▶ Considerando o problema da instalação de uma linha de televisão a cabo que conecte seis cidades do grafo ao lado, vamos assumir que você não sabe os custos de instalar a linha de TV a cabo entre todos os pares de cidades e vamos escolher uma cidade para iniciar a análise.



Árvore Geradora Mínima



Começando em AJO

- ▶ Apenas duas cidade são alcançadas a partir de Ajo: Bordo e Danza.

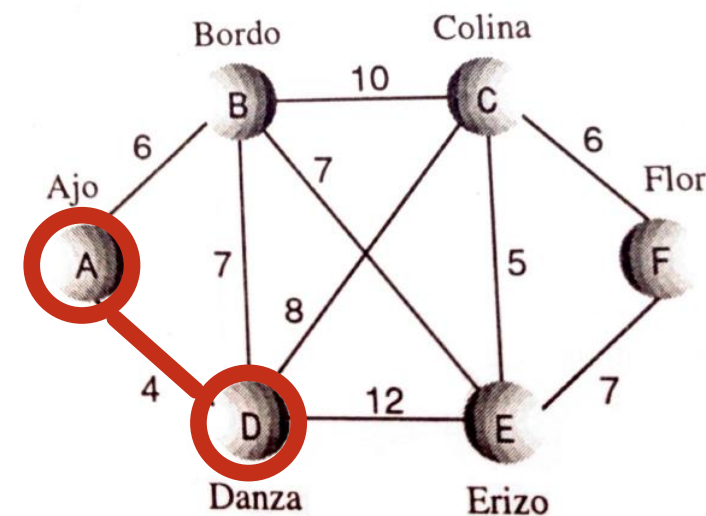
Temos os seguintes custos:

- ▶ Ajo-Danza, \$4 milhões;
- ▶ Ajo-Bordo, \$6 milhões.



Concluimos que a rota Ajo-Danza fará parte da árvore geradora mínima.

REGRA: A partir da lista, sempre selecione a aresta mais barata.



	A	B	C	D	E	F
A	inf	6	4	inf	inf	inf
B	6	inf	inf	10	7	inf
C	4	inf	inf	8	12	inf
D	inf	10	8	inf	inf	6
E	inf	7	12	inf	inf	7
F	inf	inf	inf	6	7	inf

Árvore Geradora Mínima



Construindo a ligação Ajo-Bordo

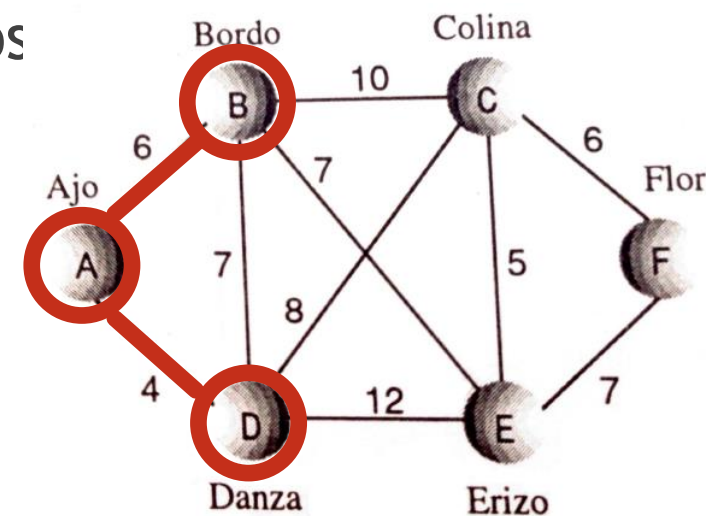
- ▶ Depois de incluir Danza na árvore geradora mínima, é necessário visitar todas as cidades não alcançáveis não visitadas a partir das cidades da árvore geradora mínima e anotar os

Temos os seguintes custos:

- ▶ **Ajo-Bordo, \$6 milhões;**
- ▶ Danza-Bordo, \$7 milhões;
- ▶ Danza-Colina, \$8 milhões;
- ▶ Danza-Erizo, \$12 milhões.



Concluimos que a rota Ajo-Bordo fará parte da árvore geradora mínima.



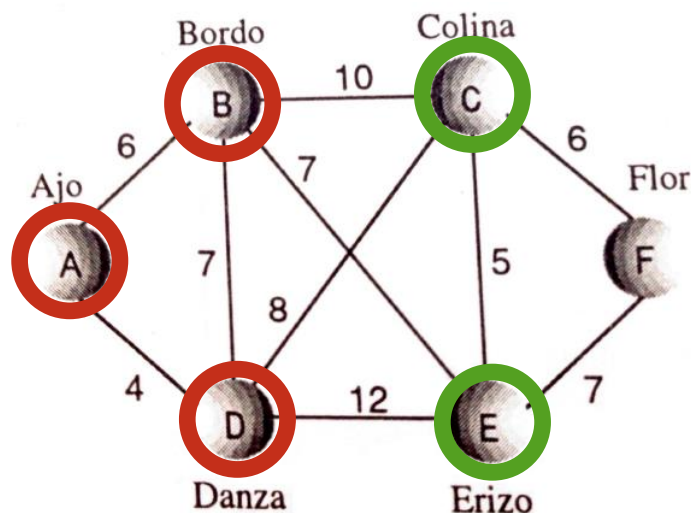
	A	B	C	D	E	F
A	inf	6	4	inf	inf	inf
B	6	inf	inf	10	7	inf
C	4	inf	inf	8	12	inf
D	inf	10	8	inf	inf	6
E	inf	7	12	inf	inf	7
F	inf	inf	inf	6	7	inf

Árvore Geradora Mínima



Em um dado momento na construção do sistema de cabos, há três tipos de cidades:

- ▶ Cidades já visitadas que já estão na árvore geradora mínima;
- ▶ Cidades que não estejam ligadas ainda e não estão na árvore geradora mínima, mas para as quais você já conhece o custo para ligá-las a pelo menos uma cidade da árvore geradora mínima.
- ▶ Cidades sobre as quais você não tenha informação.



Anotações de Custo

- ▶ Danza-Corina, \$8 milhões;
- ▶ Danza-Erizo, \$12 milhões.

Árvore Geradora Mínima



Construindo a ligação Bordo-Erizo

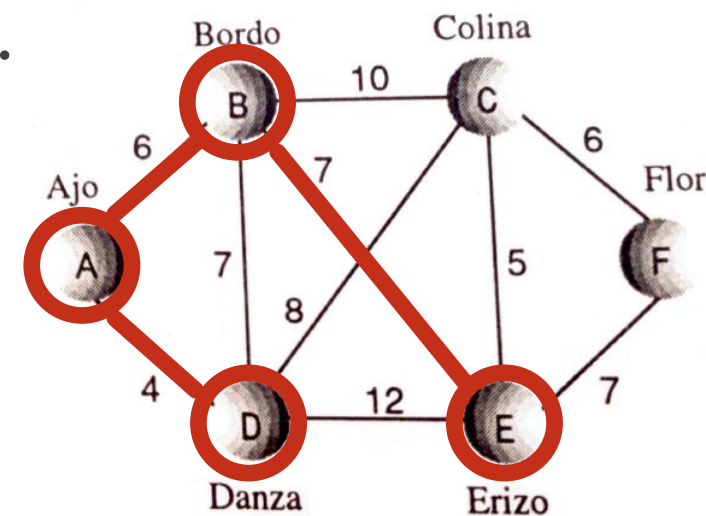
- ▶ Depois de incluir Bordo na árvore geradora mínima, é necessário visitar todas as cidades alcançáveis não visitadas a partir das cidades da árvore geradora mínima e anotar os custos.

Temos os seguintes custos:

- ▶ **Bordo-Erizo, \$7 milhões;**
- ▶ Danza-Corina, \$8 milhões;
- ▶ Bordo-Corina, \$10 milhões;
- ▶ Danza-Erizo, \$12 milhões.



Concluimos que a rota Bordo-Erizo fará parte da árvore geradora mínima.



	A	B	C	D	E	F
A	inf	6	4	inf	inf	inf
B	6	inf	inf	10	7	inf
C	4	inf	inf	8	12	inf
D	inf	10	8	inf	inf	6
E	inf	7	12	inf	inf	7
F	inf	inf	inf	6	7	inf

Árvore Geradora Mínima



Construindo a ligação Erizo-Corina

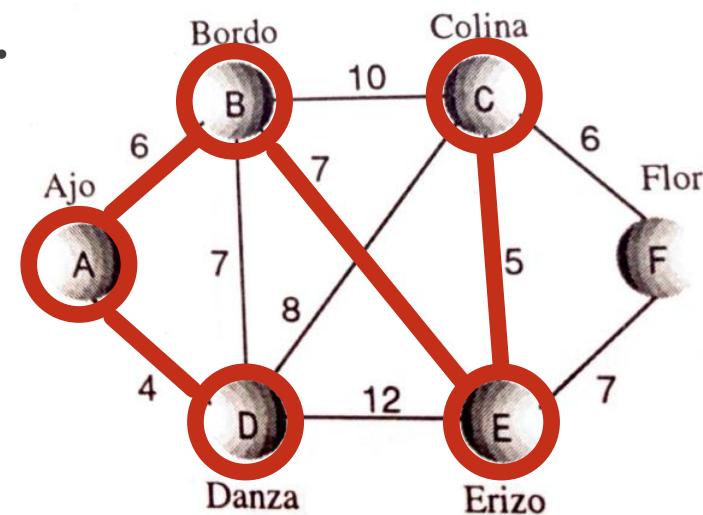
- ▶ Depois de incluir Erizo na árvore geradora mínima, é necessário visitar todas as cidades alcançáveis não visitadas a partir das cidades da árvore geradora mínima e anotar os custos.

Temos os seguintes custos:

- ▶ **Erizo-Corina, \$5 milhões;**
- ▶ Erizo-Flor, \$7 milhões;
- ▶ Danza-Corina, \$8 milhões;
- ▶ Bordo-Corina, \$10 milhões.



Concluimos que a rota Erizo-Corina fará parte da árvore geradora mínima.



	A	B	C	D	E	F
A	inf	6	4	inf	inf	inf
B	6	inf	inf	10	7	inf
C	4	inf	inf	8	12	inf
D	inf	10	8	inf	inf	6
E	inf	7	12	inf	inf	7
F	inf	inf	inf	6	7	inf

Árvore Geradora Mínima



Construindo a ligação Corina-Flor

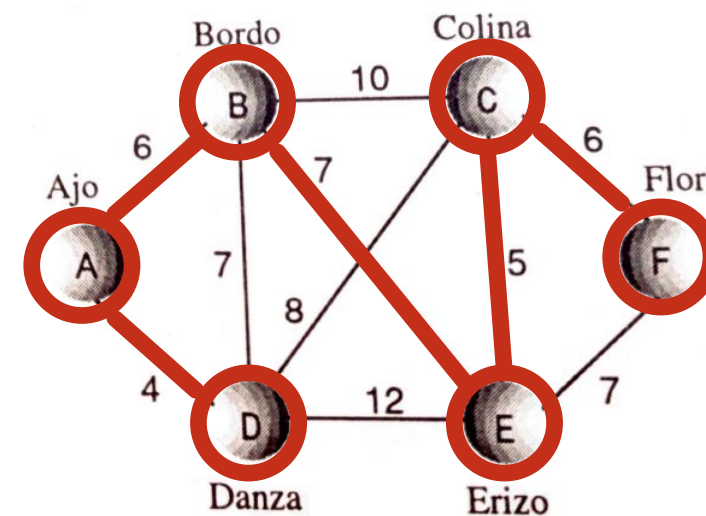
- ▶ As opções estão acabando. Depois de remover cidades já ligadas, suas lista agora mostrará apenas os custos abaixo.

Temos os seguintes custos:

- ▶ **Corina-Flor, \$6 milhões;**
- ▶ Erizo-Flor, \$7 milhões.



Concluimos que a rota Corina-Flor fará parte da árvore geradora mínima.



	A	B	C	D	E	F
A	inf	6	4	inf	inf	inf
B	6	inf	inf	10	7	inf
C	4	inf	inf	8	12	inf
D	inf	10	8	inf	inf	6
E	inf	7	12	inf	inf	7
F	inf	inf	inf	6	7	inf

Árvore Geradora Mínima



Esboço do Algoritmo em Terminologia de Grafo

Comece com um nó e coloque-o na árvore. Então, repetidamente, faça o seguinte:

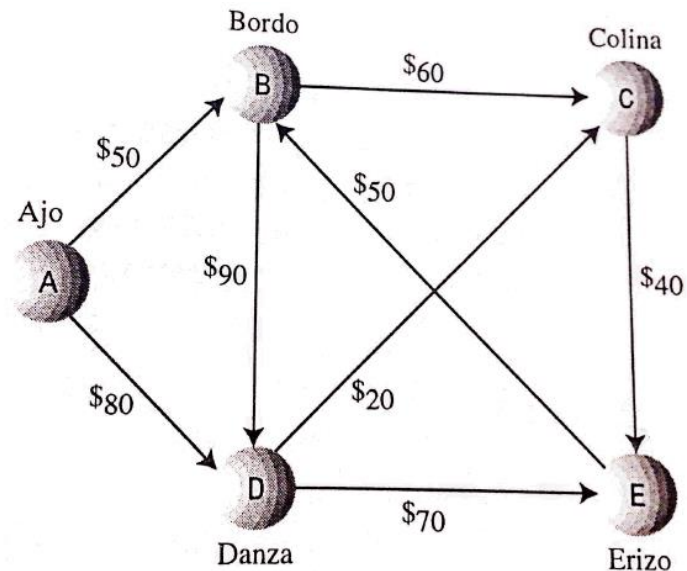
- ▶ Localize todas as arestas a partir do nó mais recente até outros nós que não estejam na árvore. Coloque essas arestas na fila de prioridade.
- ▶ Selecione a aresta com o menor peso e adicione essa aresta e seu nó de destino à árvore geradora mínima.

O Problema do Caminho mais Curto



O problema mais comumente encontrado associado a grafos ponderados é localizar o caminho mais curto entre dois nós.

- ▶ Como exemplo, dessa vez estamos preocupados com ferrovias em vez de TV a cabo.
- ▶ Não iremos construir a ferrovia, ela já existe. Apenas queremos encontrar a rota mais barata de uma cidade para outra.



	A	B	C	D	E
A	inf	50	inf	80	inf
B	inf	inf	60	90	inf
C	inf	inf	inf	inf	40
D	inf	inf	20	inf	70
E	inf	50	inf	inf	inf

Algoritmo de Dijkstra



Uma solução para o problema do caminho mais curto é o algoritmo de Dijkstra.

- ▶ Este algoritmo não só encontra caminho mais curto a partir de um nó especificado até outro, como também os caminhos mais curtos do nó especificado até todos os outros.
- ▶ O algoritmo tem que ver parte da informação por vez, portanto, como no algoritmo anterior, iremos supor que você é do mesmo modo incapaz de ver o todo.

Algoritmo de Dijkstra

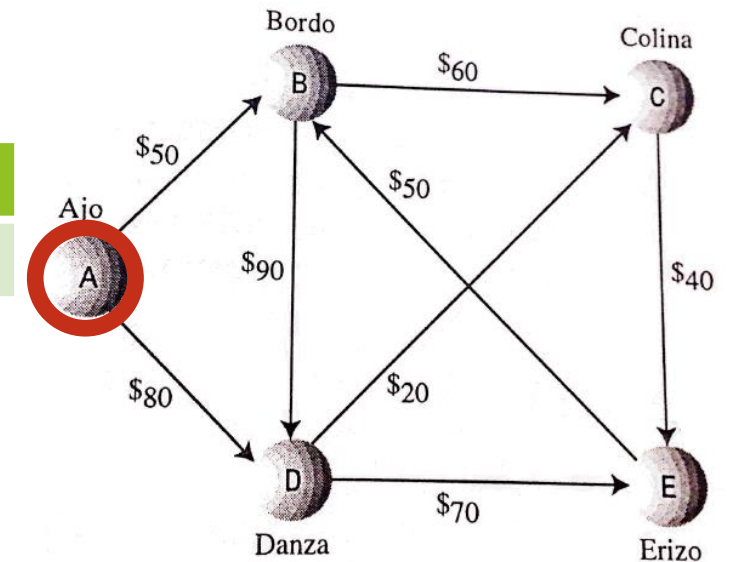


Visitando Ajo

- ▶ Você deve visitar cada cidade, obtendo informações sobre custos para outras cidades e anotando na tabela.

De Ajo para ->	Bordo	Corina	Danza	Erizo
Passo 1	50 (via A)	inf	80 (via A)	Inf

REGRA: Sempre visite outra cidade cujo preço total a partir do ponto de origem seja o mais barato



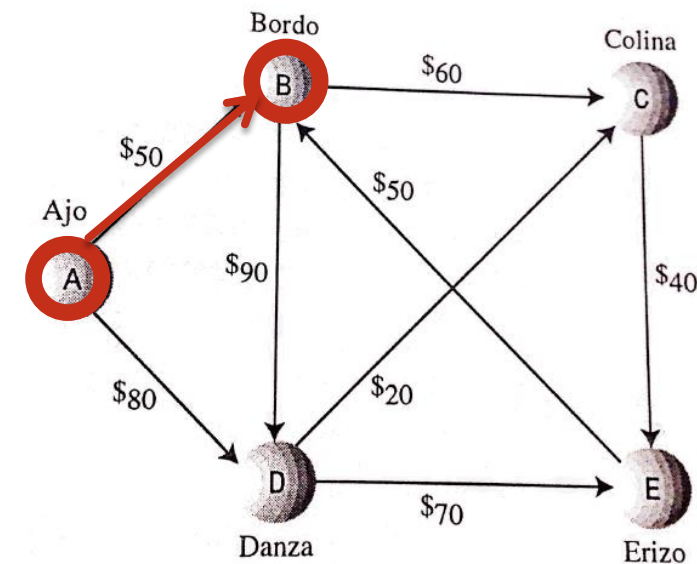
	A	B	C	D	E
A	inf	50	inf	80	inf
B	inf	inf	60	90	inf
C	inf	inf	inf	inf	40
D	inf	inf	20	inf	70
E	inf	50	inf	inf	inf

Algoritmo de Dijkstra



Visitando Bordo

- ▶ A tarifa mais barata de Ajo para Bordo é 50. Quando visitar Bordo, é possível ter a informação que custa \$60 para viajar para Corina e \$90 para Danza.
- ▶ Você calcula que tem que ser \$110 (\$50 mais \$60) para ir de Ajo para Corina através de Bordo.
- ▶ Porém, neste ponto você já sabe que custa apenas \$80 para ir diretamente de Ajo para Danza.



De Ajo para ->	Bordo	Corina	Danza	Erizo
Passo 1	50 (via A)	inf	80 (via Ajo)	Inf
Passo 2	50 (via Ajo) *	110 (via Bordo)	80 (via Ajo)	

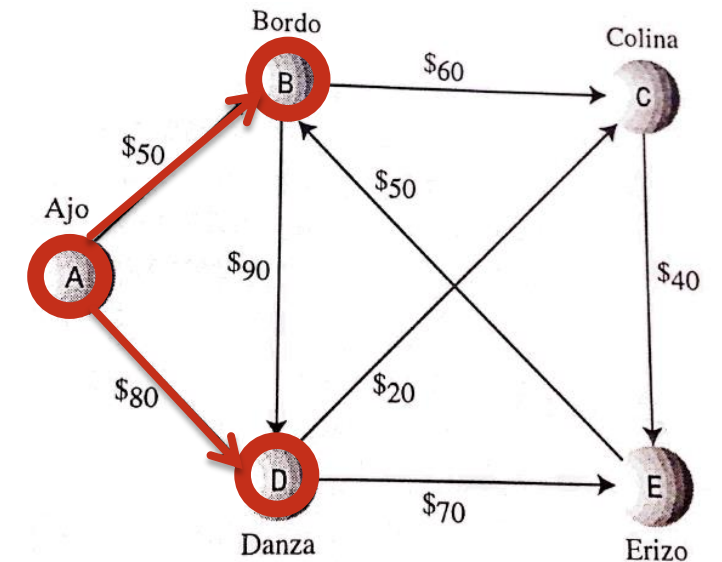
	A	B	C	D	E
A	inf	50	inf	80	inf
B	inf	inf	60	90	inf
C	inf	inf	inf	inf	40
D	inf	inf	20	inf	70
E	inf	50	inf	inf	inf

Algoritmo de Dijkstra



Visitando Danza

De Ajo para ->	Bordo	Corina	Danza	Erizo
Passo 1	50 (via A)	inf	80 (via Ajo)	Inf
Passo 2	50 (via Ajo) *	110 (via Bordo)	80 (via Ajo)	Inf
Passo 3	50 (via Ajo) *	100 (via Danza)	80 (via Ajo) *	150 (via Danza)



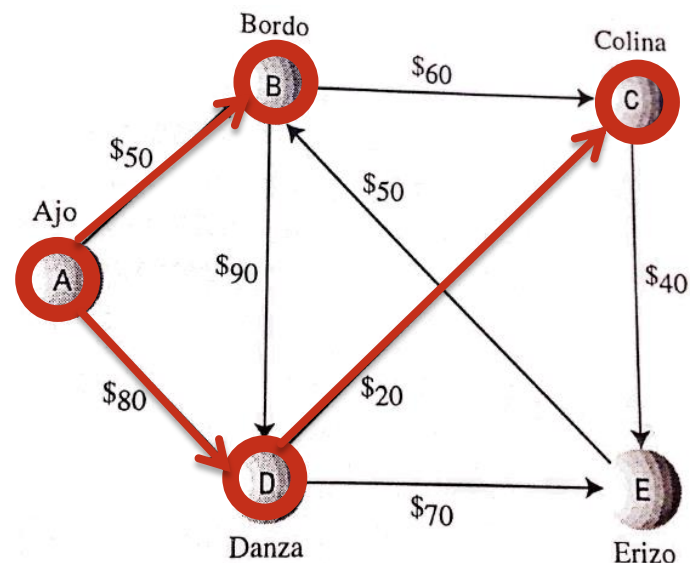
	A	B	C	D	E
A	inf	50	inf	80	inf
B	inf	inf	60	90	inf
C	inf	inf	inf	inf	40
D	inf	inf	20	inf	70
E	inf	50	inf	inf	inf

Algoritmo de Dijkstra



Visitando Colina

De Ajo para ->	Bordo	Corina	Danza	Erizo
Passo 1	50 (via A)	inf	80 (via Ajo)	Inf
Passo 2	50 (via Ajo) *	110 (via Bordo)	80 (via Ajo)	Inf
Passo 3	50 (via Ajo) *	100 (via Bordo)	80 (via Ajo) *	150 (via Danza)
Passo 4	50 (via Ajo) *	100 (via Danza) *	80 (via Ajo) *	140 (via Colina)



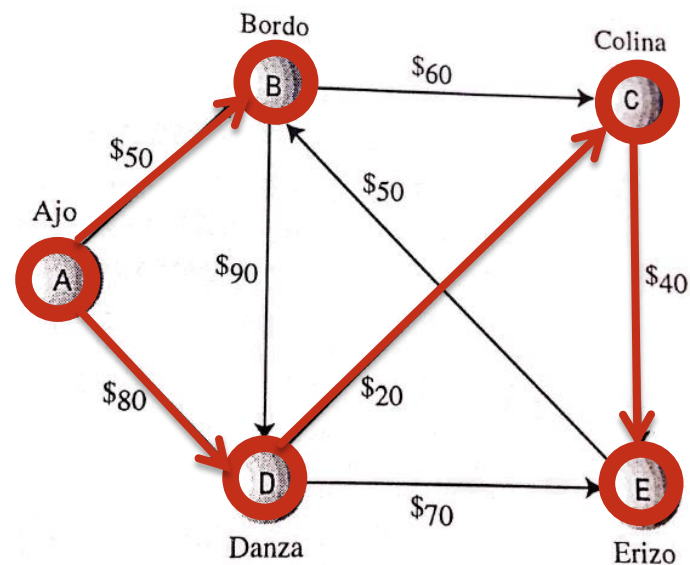
	A	B	C	D	E
A	inf	50	inf	80	inf
B	inf	inf	60	90	inf
C	inf	inf	inf	inf	40
D	inf	inf	20	inf	70
E	inf	50	inf	inf	inf

Algoritmo de Dijkstra



Visitando Erizo

De Ajo para ->	Bordo	Corina	Danza	Erizo
Passo 1	50 (via A)	inf	80 (via Ajo)	Inf
Passo 2	50 (via Ajo) *	110 (via Bordo)	80 (via Ajo)	Inf
Passo 3	50 (via Ajo) *	100 (via Bordo)	80 (via Ajo) *	150 (via Danza)
Passo 4	50 (via Ajo) *	100 (via Danza) *	80 (via Ajo) *	140 (via Colina)
Passo 5	50 (via Ajo) *	100 (via Danza) *	80 (via Ajo) *	140 (via Colina) *



	A	B	C	D	E
A	inf	50	inf	80	inf
B	inf	inf	60	90	inf
C	inf	inf	inf	inf	40
D	inf	inf	20	inf	70
E	inf	50	inf	inf	inf

Algoritmo de Dijkstra



Essa narrativa demonstrou o essencial do algoritmo de Dijkstra. Os pontos chave são:

- ▶ A cada vez que você visitar uma nova cidade, usa as novas informações fornecidas para revisar a lista de tarifas. Apenas a tarifa mais barata (que você conhece) do ponto inicial até uma dada cidade é mantida.
- ▶ Você sempre visita uma nova cidade que tenha o caminho mais barato a partir do ponto inicial (não a aresta mais barata de qualquer cidade).