

Algoritmos de Dijkstra e Bellman-Ford

Prof. Celso A. W. Santos

J702 :: Teoria de Grafos

celso.santos@docente.unip.br

24/04/2020

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:
[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!
Exemplo: [J702 - Lista 3] Matrícula D648H12
 - ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!
Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:
[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!
Exemplo: [J702 - Lista 3] Matrícula D648H12
 - ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!
Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:

[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!

Exemplo: [J702 - Lista 3] Matrícula D648H12

- ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!
Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:

[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!

Exemplo: [J702 - Lista 3] Matrícula D648H12

- ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!
Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:

[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!

Exemplo: [J702 - Lista 3] Matrícula D648H12

- ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!

Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:

[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!

Exemplo: [J702 - Lista 3] Matrícula D648H12

- ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!

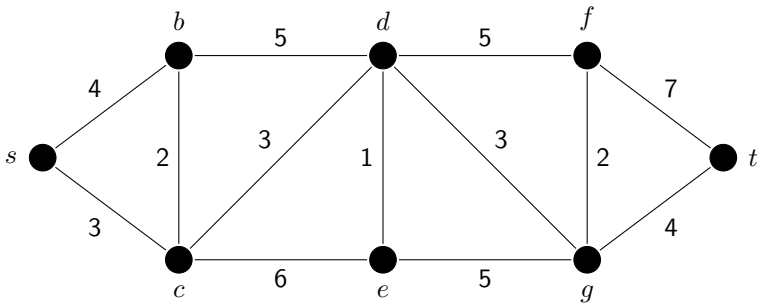
Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

- ◇ Avaliações serão realizadas por **listas de exercícios**, disponibilizadas em meu site todas as sextas-feiras.
 - ▷ Prazo de entrega: até a **quinta**-feira da semana seguinte, às **23h59m**
 - ▷ Submissão deve ser feita em formato **PDF**
 - ▷ Enviar com assunto no e-mail:
[J702 - Lista X] Matrícula XXXXXXXX ← tudo maiúsculo!

Exemplo: [J702 - Lista 3] Matrícula D648H12
 - ▷ Sim... eu vou zerar estudos submetidos fora do padrão :)
- ◇ Todas as listas deverão ser entregues *presencialmente* ao retorno das atividades!
Se você não tem impressora...
- ◇ As notas das Listas 2 e 3 estarão disponíveis no site na sexta-feira que vem, dia 01/05.

Na aula passada...

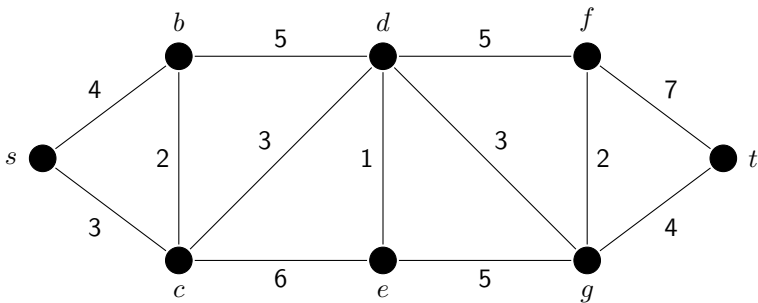
Definição. Um **grafo ponderado** é uma tripla $G = (V, U, w)$ tal que V é um conjunto de vértices, E é um conjunto de arestas, e $w : E \rightarrow \mathbb{R}$ é uma função **peso** que atribui um valor real a cada aresta do grafo.



CAMINHO MÍNIMO

Entrada: Um grafo ponderado $G = (V, U, w)$, um vértice fonte $s \in V$ e um vértice destino $t \in V$.

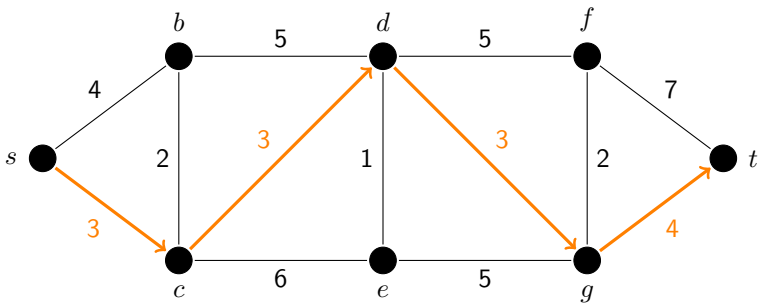
Pergunta: Qual é o menor caminho entre s e t ? E qual é o seu custo?



CAMINHO MÍNIMO

Entrada: Um grafo ponderado $G = (V, U, w)$, um vértice fonte $s \in V$ e um vértice destino $t \in V$.

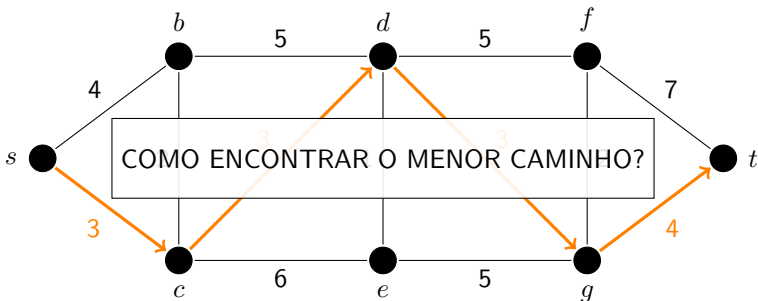
Pergunta: Qual é o menor caminho entre s e t ? E qual é o seu custo?



CAMINHO MÍNIMO

Entrada: Um grafo ponderado $G = (V, U, w)$, um vértice fonte $s \in V$ e um vértice destino $t \in V$.

Pergunta: Qual é o menor caminho entre s e t ? E qual é o seu custo?



Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ "Resolve" o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ "Resolve" o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ "Resolve" o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ "Resolve" o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ "Resolve" o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ “Resolve” o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ “Resolve” o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ “Resolve” o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ “Resolve” o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ “Resolve” o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Dados vértices $s, t \in V$, encontra o menor caminho entre s e t .
- ▷ Tem um problema!

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ “Resolve” o problema!

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Mais rápido que executar *Single-source* para todos os vértices

Algoritmo de Dijkstra

INICIALIZA-SINGLE-SOURCE(G, s):

para todo vértice $v \in V$:

$d[v] = \infty$;

$\pi[v] = \perp$;

$d[s] = 0$;

RELAXA(u, v, w):

se $d[v] > d[u] + w(u, v)$:

$d[v] = d[u] + w(u, v)$;

$\pi[v] = u$;

DIJKSTRA(G, s):

 INICIALIZA-SINGLE-SOURCE(G, s)

$S = \emptyset$;

$Q = V$;

enquanto $Q \neq \emptyset$ **faça**:

$u = \text{EXTRAIR-MIN}(Q)$;

$S = S \cup \{u\}$;

p/ todo vértice $v \in N(u)$ **faça**:

 RELAXA(u, v, w);

INICIALIZA-SINGLE-SOURCE(G, s):

para todo vértice $v \in V$:

$d[v] = \infty$;

$\pi[v] = \perp$;

$d[s] = 0$;

RELAXA(u, v, w):

se $d[v] > d[u] + w(u, v)$:

$d[v] = d[u] + w(u, v)$;

$\pi[v] = u$;

DIJKSTRA(G, s):

 INICIALIZA-SINGLE-SOURCE(G, s)

$S = \emptyset$;

$Q = V$;

enquanto $Q \neq \emptyset$ **faça**:

$u = \text{EXTRAIR-MIN}(Q)$;

$S = S \cup \{u\}$;

p/ todo vértice $v \in N(u)$ **faça**:

 RELAXA(u, v, w);

INICIALIZA-SINGLE-SOURCE(G, s):

para todo vértice $v \in V$:

$d[v] = \infty$;

$\pi[v] = \perp$;

$d[s] = 0$;

RELAXA(u, v, w):

se $d[v] > d[u] + w(u, v)$:

$d[v] = d[u] + w(u, v)$;

$\pi[v] = u$;

DIJKSTRA(G, s):

 INICIALIZA-SINGLE-SOURCE(G, s)

$S = \emptyset$;

$Q = V$;

enquanto $Q \neq \emptyset$ **faça**:

$u = \text{EXTRAIR-MIN}(Q)$;

$S = S \cup \{u\}$;

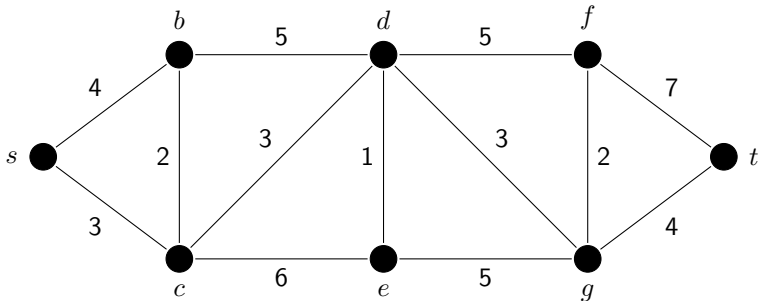
p/ todo vértice $v \in N(u)$ **faça**:

 RELAXA(u, v, w);

Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>								

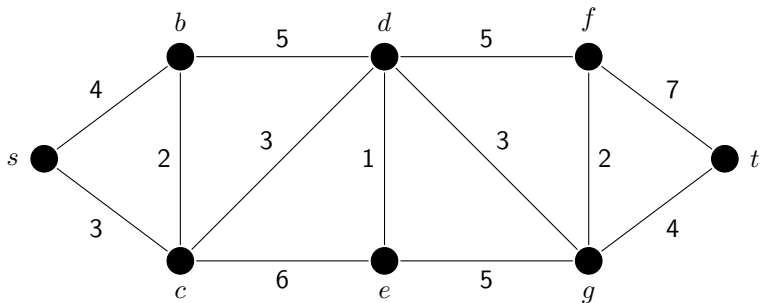
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π								



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	∞	∞	∞	∞	∞	∞	∞

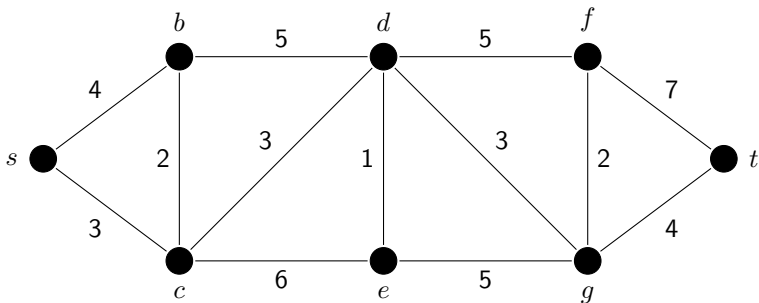
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	\perp	\perp	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	∞	∞	∞	∞	∞	∞	∞

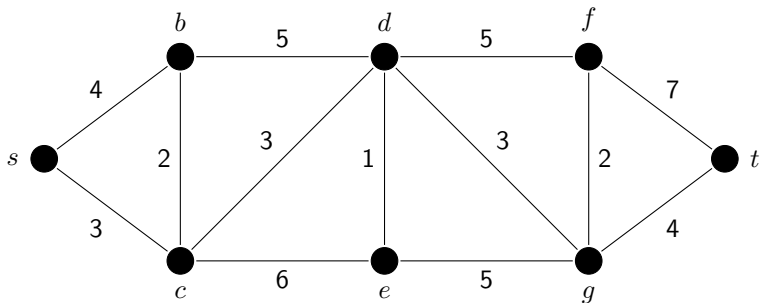
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	\perp	\perp	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	∞	∞	∞	∞	∞	∞

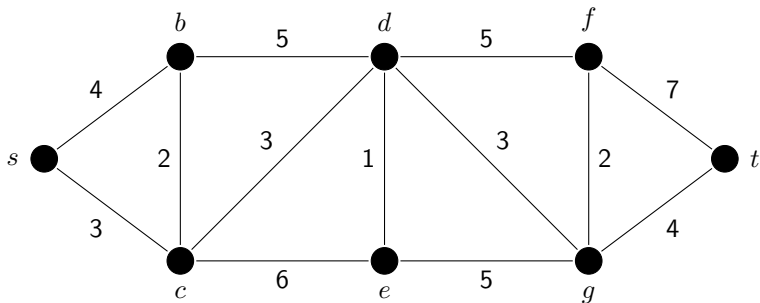
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	\perp	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	∞	∞	∞	∞	∞

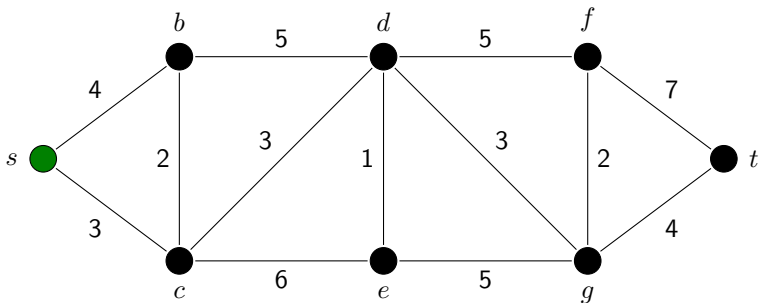
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	∞	∞	∞	∞	∞

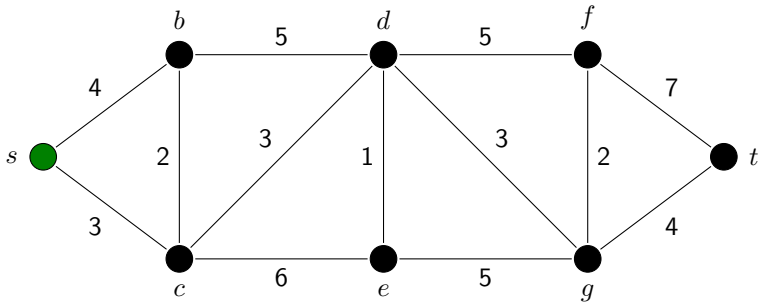
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	∞	∞	∞	∞	∞

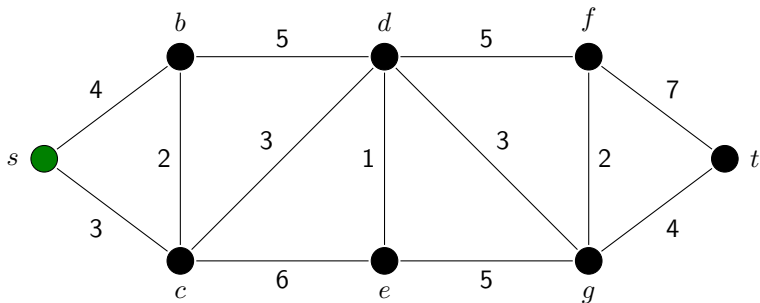
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	∞	∞	∞	∞

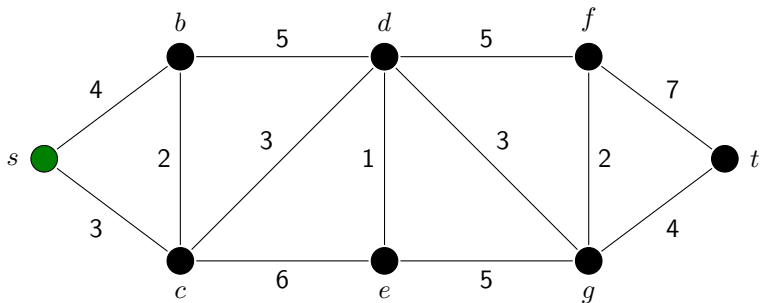
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	9	∞	∞	∞

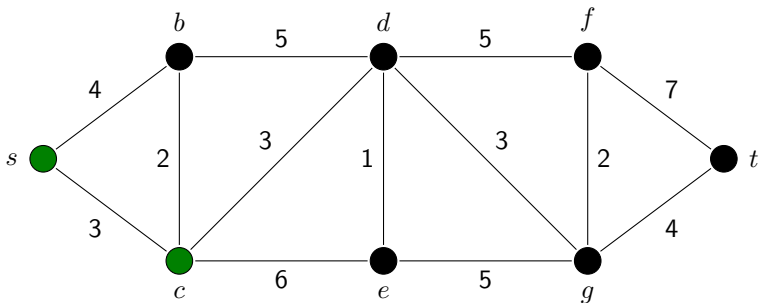
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	9	∞	∞	∞

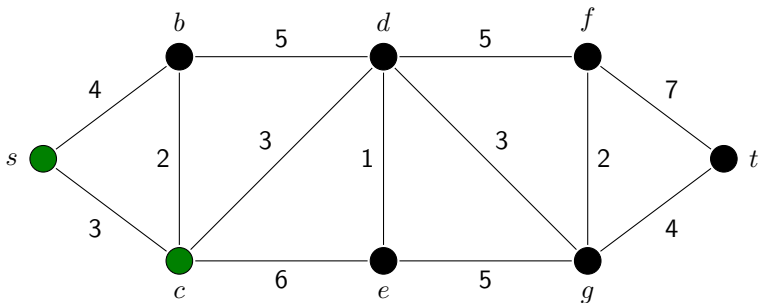
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	9	∞	∞	∞

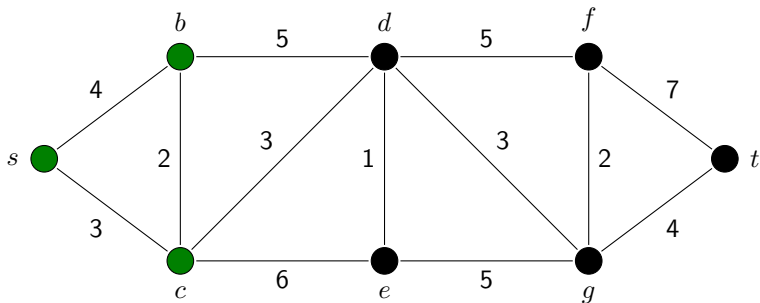
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	9	∞	∞	∞

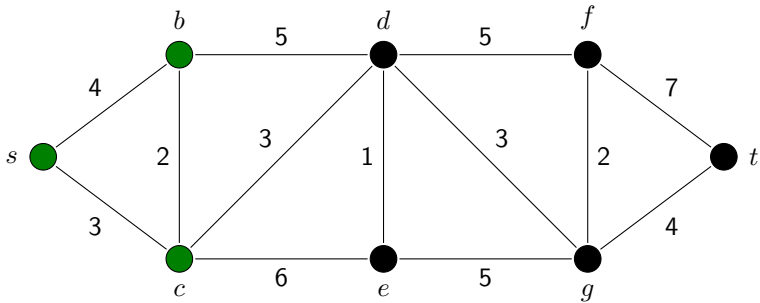
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	9	∞	∞	∞

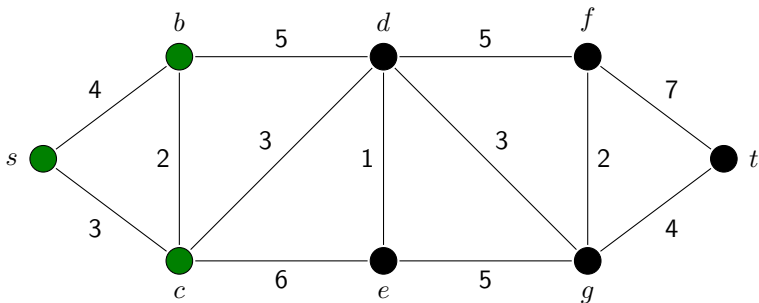
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	∞	∞	∞

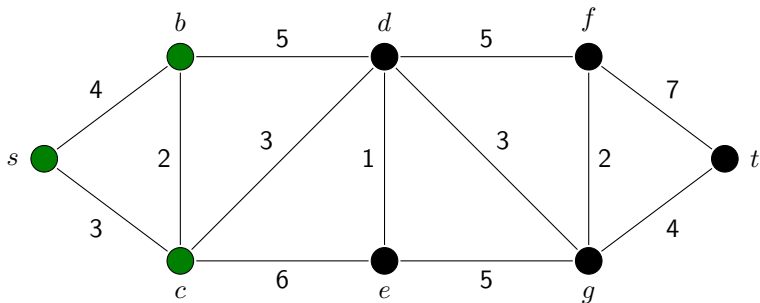
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	∞	∞

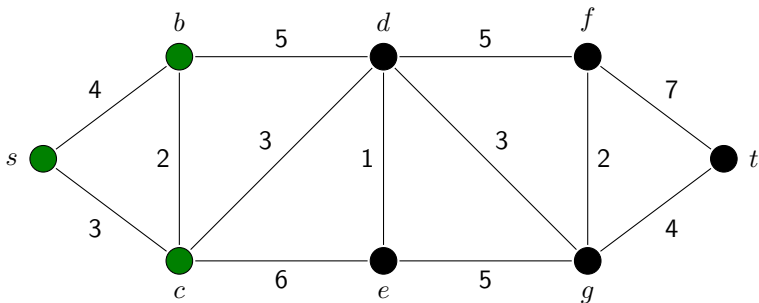
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	\perp	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	∞

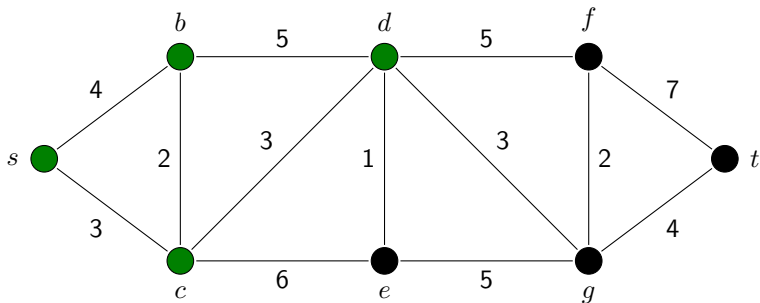
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	∞

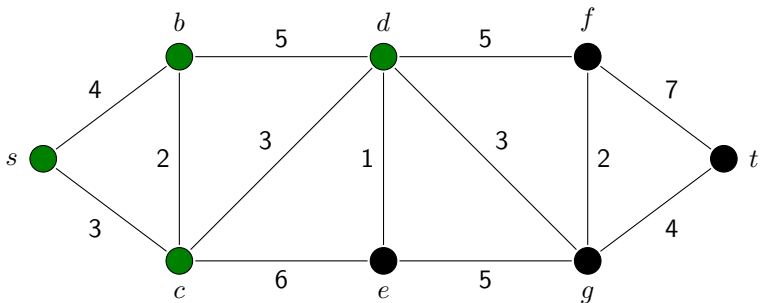
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	∞

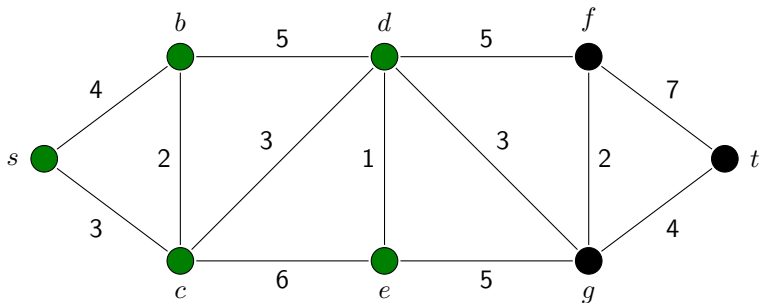
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	∞

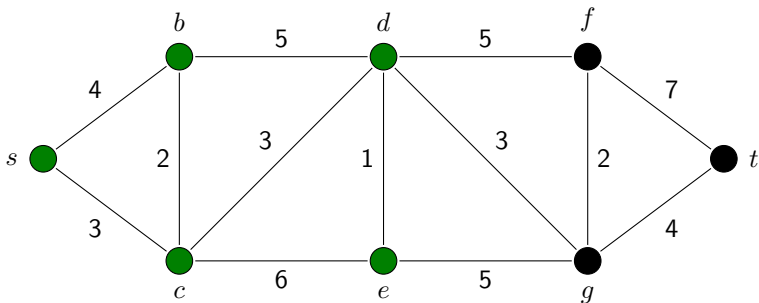
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	∞

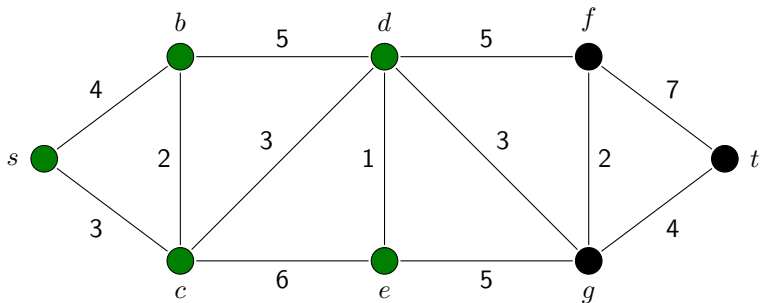
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	\perp



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

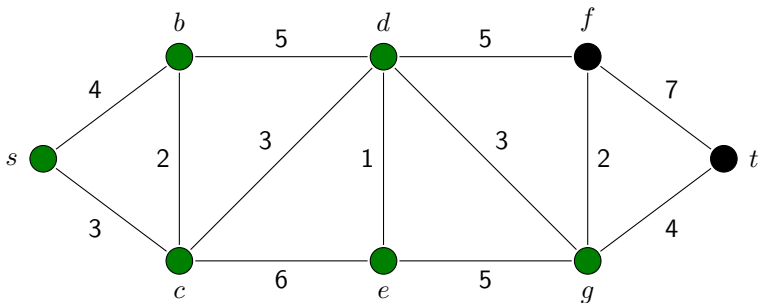
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

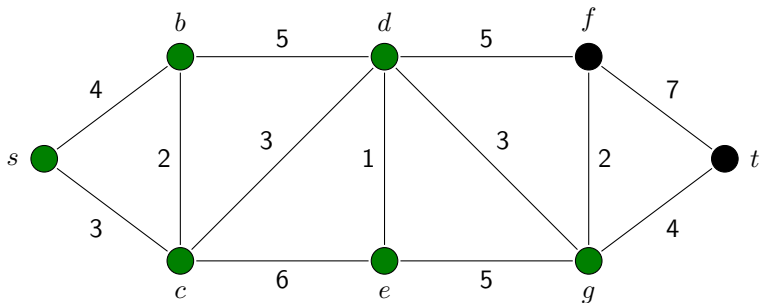
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

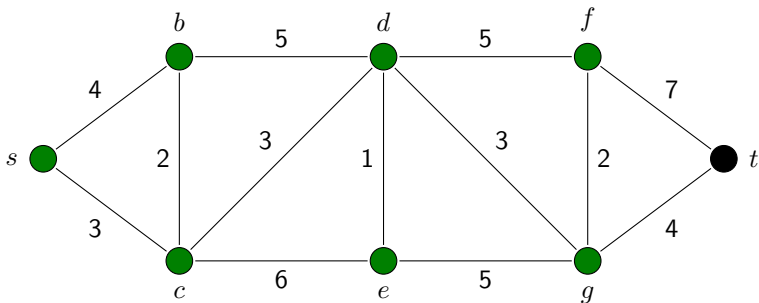
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

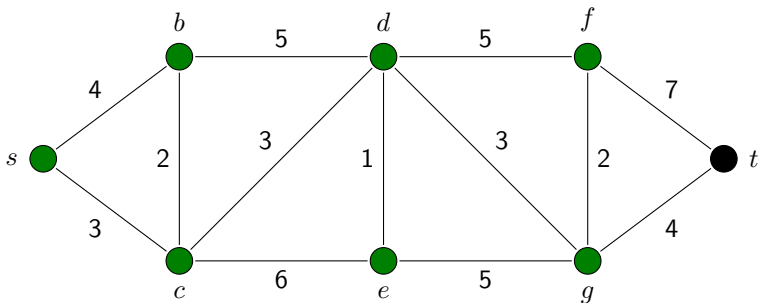
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

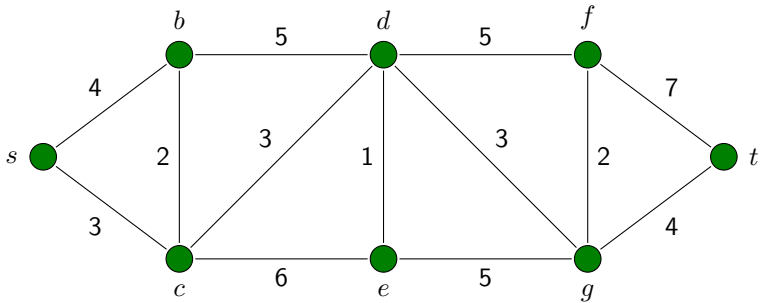
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

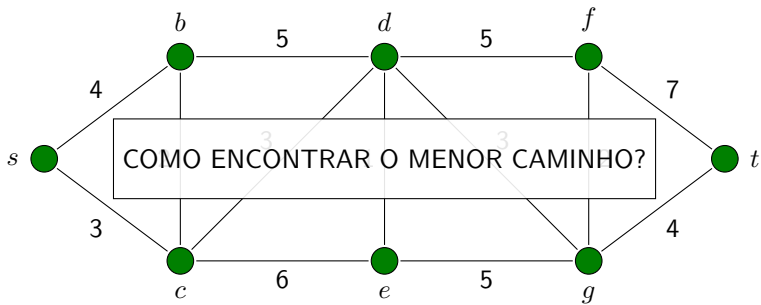
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

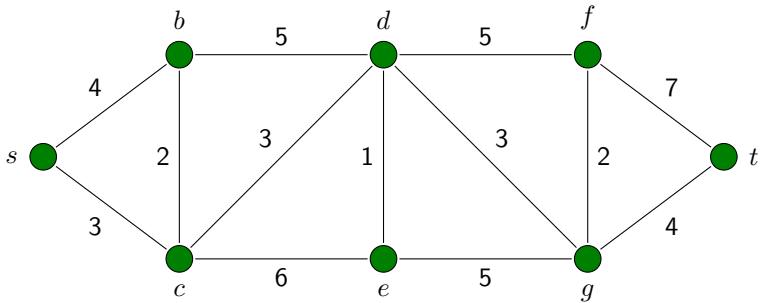
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

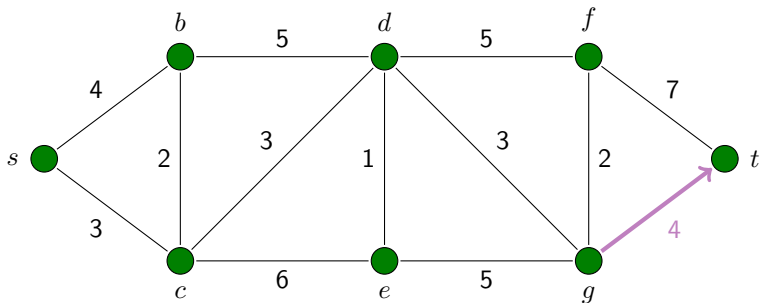
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

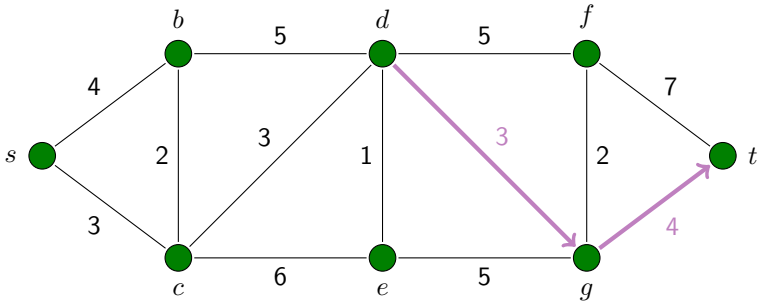
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	s	b	c	d	e	f	g	t
d	0	4	3	6	7	11	9	13

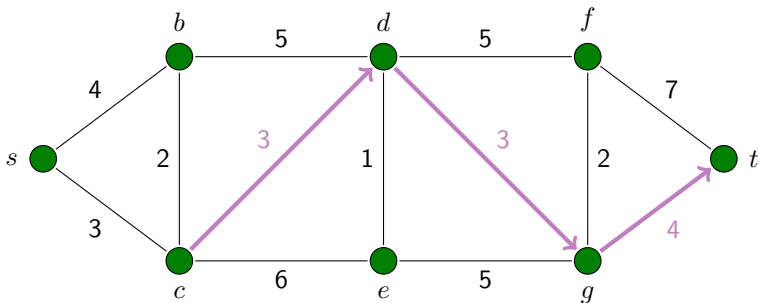
	s	b	c	d	e	f	g	t
π	\perp	s	s	c	d	d	d	g



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

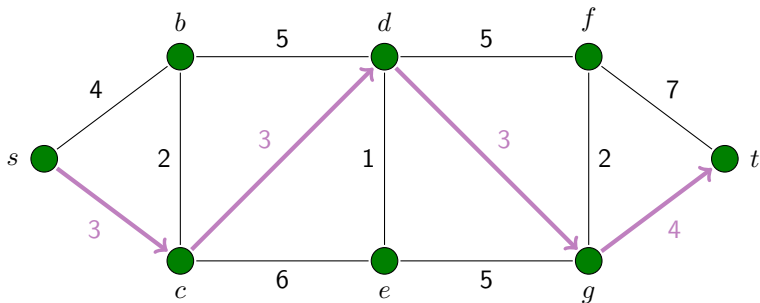
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

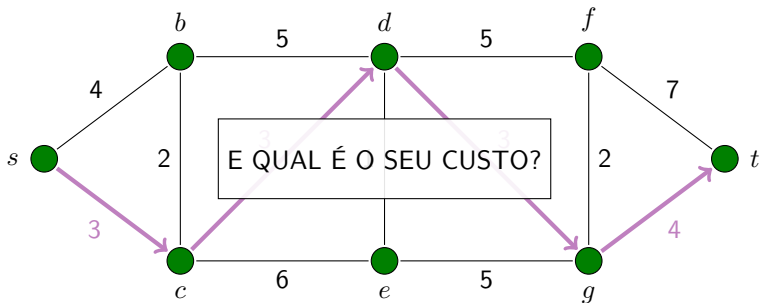
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

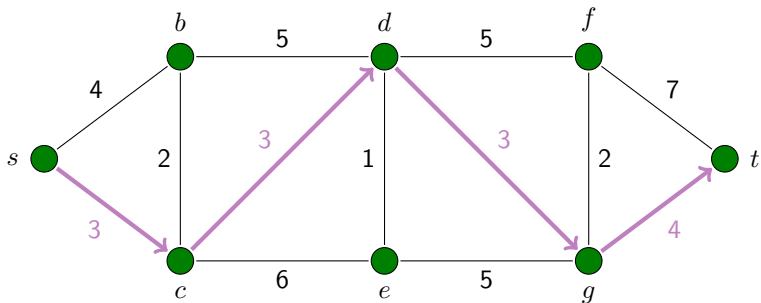
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



Exemplo de Execução :: Algoritmo de Dijkstra

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
<i>d</i>	0	4	3	6	7	11	9	13

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>t</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>	<i>d</i>	<i>d</i>	<i>g</i>



DEVOLVE-MENOR-CAMINHO(G, s):

DIJKSTRA(G, s);

imprime "Custo: " + $d[t]$;

imprime "Caminho: $t \leftarrow$ "

$v = \pi[t]$;

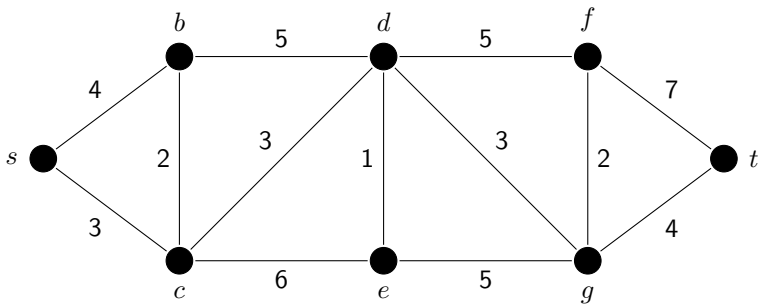
enquanto $v \neq \perp$ **faça**:

imprime: " $v \leftarrow$ ";

$v = \pi[v]$;

O problema do Dijkstra: Ciclos Negativos

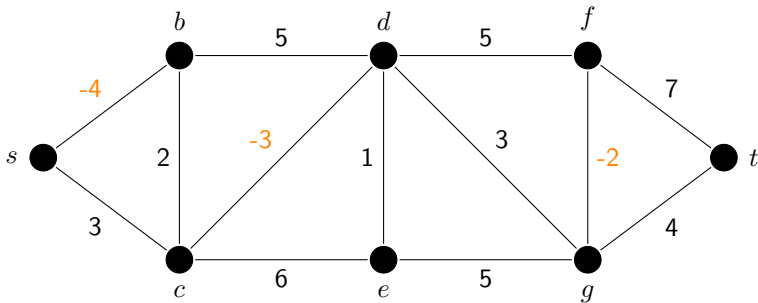
- ◇ O algoritmo de Dijkstra funciona mesmo quando existem **pesos negativos** nas arestas.



- ◇ ... mas quebra quando existem **ciclos negativos**!

O problema do Dijkstra: Ciclos Negativos

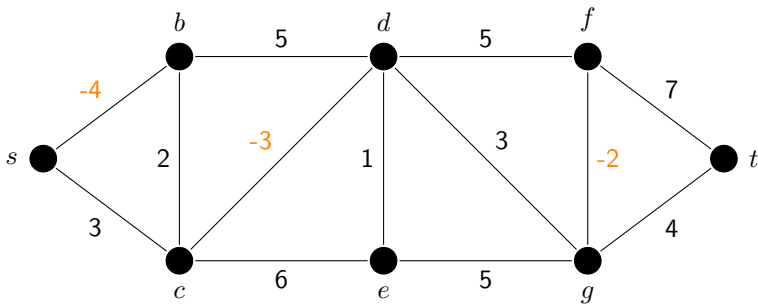
- ◇ O algoritmo de Dijkstra funciona mesmo quando existem **pesos negativos** nas arestas.



- ◇ ... mas quebra quando existem **ciclos negativos**!

O problema do Dijkstra: Ciclos Negativos

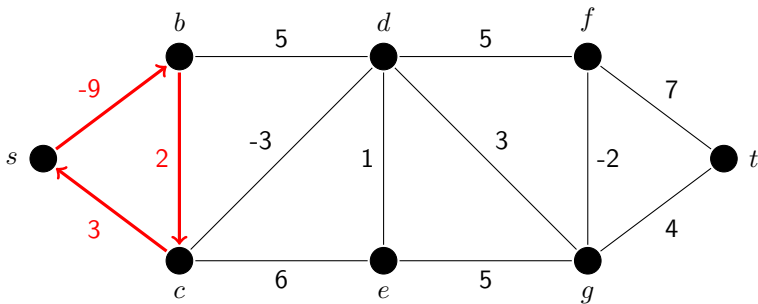
- ◇ O algoritmo de Dijkstra funciona mesmo quando existem **pesos negativos** nas arestas.



- ◇ ... mas quebra quando existem **ciclos negativos**!

O problema do Dijkstra: Ciclos Negativos

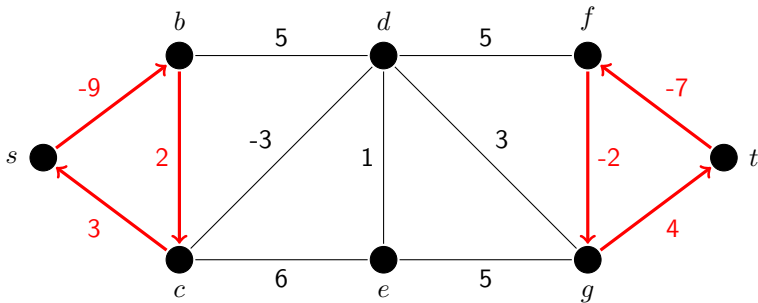
- ◇ O algoritmo de Dijkstra funciona mesmo quando existem **pesos negativos** nas arestas.



- ◇ ... mas quebra quando existem **ciclos negativos**!

O problema do Dijkstra: Ciclos Negativos

- ◇ O algoritmo de Dijkstra funciona mesmo quando existem **pesos negativos** nas arestas.



- ◇ ... mas quebra quando existem **ciclos negativos**!

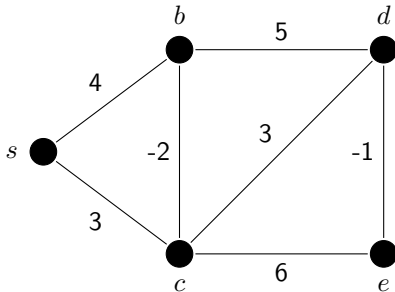
O Algoritmo de Bellman-Ford

```
BELLMAN-FORD( $G, s$ ):  
  INICIALIZA-SINGLE-SOURCE( $G, s$ );  
  de  $i = 1$  até  $n - 1$  faça:  
    para toda aresta  $uv \in E$  faça:  
      RELAXA( $u, v, w$ );  
  para toda aresta  $uv \in E$  faça:  
    se  $d[v] > d[u] + w(u, v)$  então:  
      devolve FALSE  
  devolve TRUE
```

Exemplo de Execução :: Algoritmo de Bellman-Ford

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>d</i>					

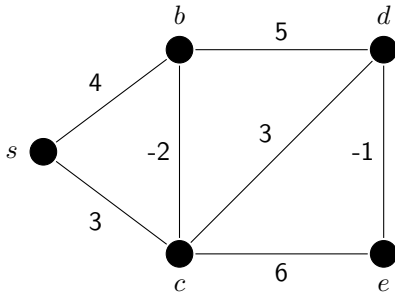
	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
π					



Exemplo de Execução :: Algoritmo de Bellman-Ford

	s	b	c	d	e
d	0	∞	∞	∞	∞

	s	b	c	d	e
π	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

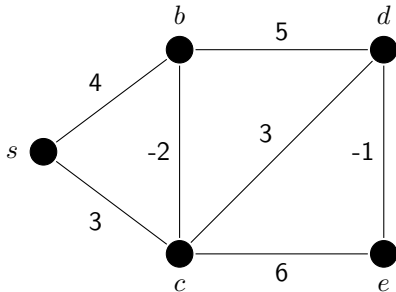
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	∞	∞	∞	∞

	s	b	c	d	e
π	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : *de*

2^a : *bd*

3^a : *sc*

4^a : *bc*

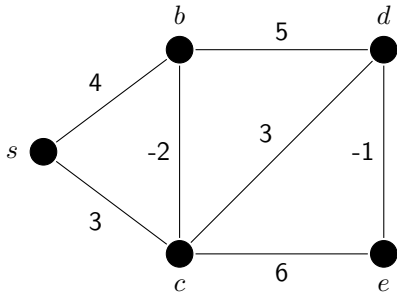
5^a : *sb*

6^a : *cd*

7^a : *ce*

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>d</i>	0	∞	∞	∞	∞

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
π	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

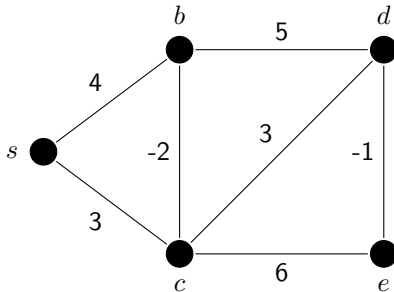
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	∞	∞	∞	∞

	s	b	c	d	e
π	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

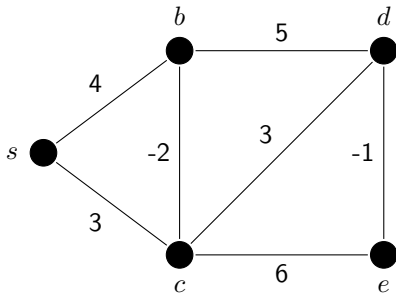
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	∞	∞	∞	∞

	s	b	c	d	e
π	\perp	\perp	\perp	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

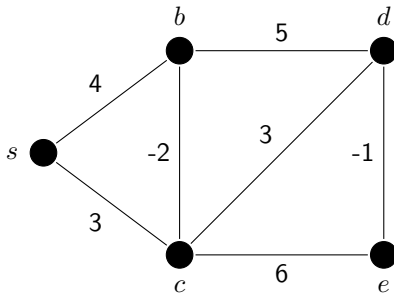
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	∞	3	∞	∞

	s	b	c	d	e
π	\perp	\perp	s	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

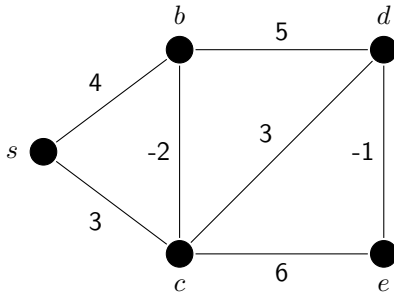
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	∞	3	∞	∞

	s	b	c	d	e
π	\perp	\perp	s	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

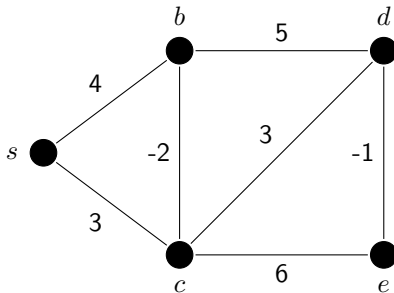
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	∞	3	∞	∞

	s	b	c	d	e
π	\perp	\perp	s	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

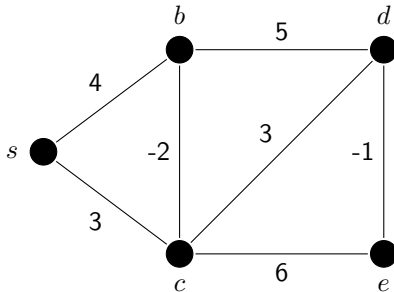
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	∞	∞

	s	b	c	d	e
π	\perp	s	s	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

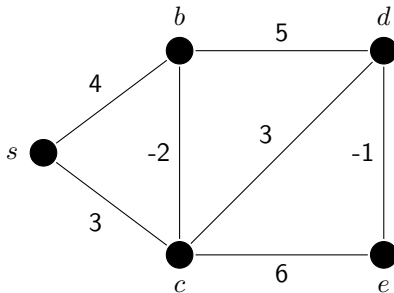
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	∞	∞

	s	b	c	d	e
π	\perp	s	s	\perp	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

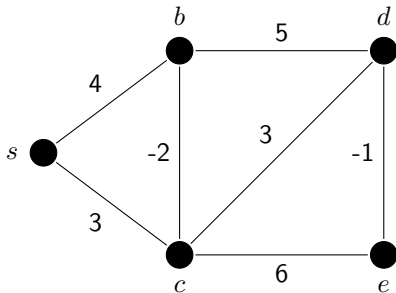
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	6	∞

	s	b	c	d	e
π	\perp	s	s	c	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

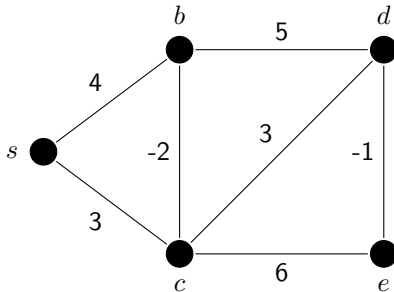
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	6	∞

	s	b	c	d	e
π	\perp	s	s	c	\perp



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

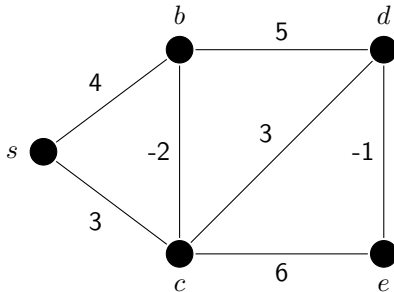
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	6	9

	s	b	c	d	e
π	\perp	s	s	c	c



Exemplo de Execução :: Algoritmo de Bellman-Ford

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>d</i>	0	4	3	6	9

Ordem das Arestas:

1^a : *de*

2^a : *bd*

3^a : *sc*

4^a : *bc*

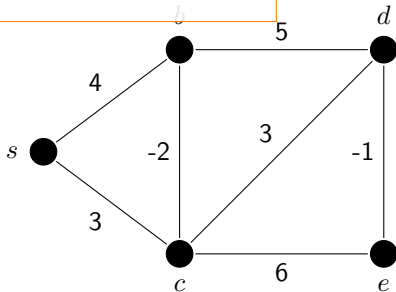
5^a : *sb*

6^a : *cd*

7^a : *ce*

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>

ACABOU O ALGORITMO?



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : *de*

2^a : *bd*

3^a : *sc*

4^a : *bc*

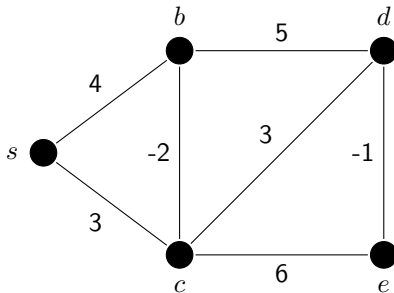
5^a : *sb*

6^a : *cd*

7^a : *ce*

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>d</i>	0	4	3	6	9

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>c</i>



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : *de*

2^a : *bd*

3^a : *sc*

4^a : *bc*

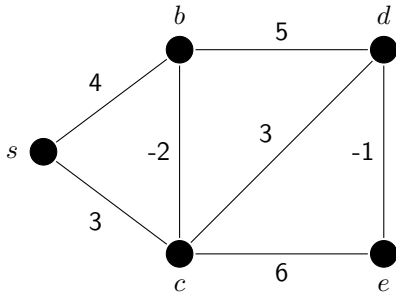
5^a : *sb*

6^a : *cd*

7^a : *ce*

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>d</i>	0	4	3	6	<i>5</i>

	<i>s</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
π	\perp	<i>s</i>	<i>s</i>	<i>c</i>	<i>d</i>



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

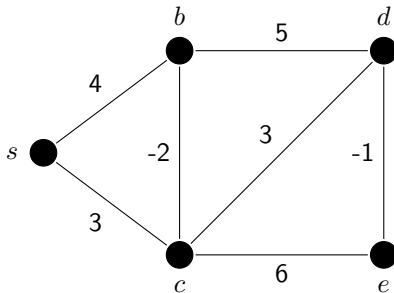
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	6	5

	s	b	c	d	e
π	\perp	s	s	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

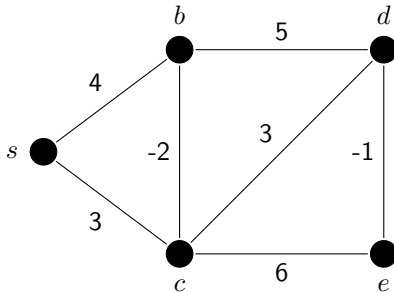
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	6	5

	s	b	c	d	e
π	\perp	s	s	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

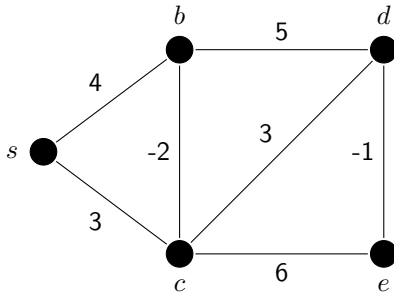
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	3	6	5

	s	b	c	d	e
π	\perp	s	s	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

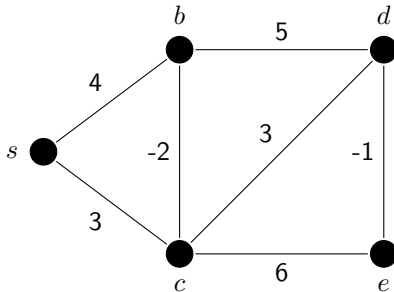
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	2	6	5

	s	b	c	d	e
π	\perp	s	b	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

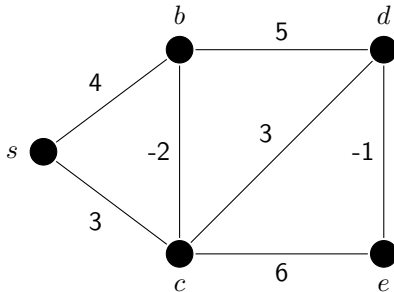
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	2	6	5

	s	b	c	d	e
π	\perp	s	b	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

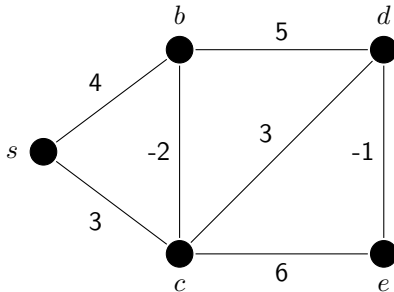
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	2	6	5

	s	b	c	d	e
π	\perp	s	b	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

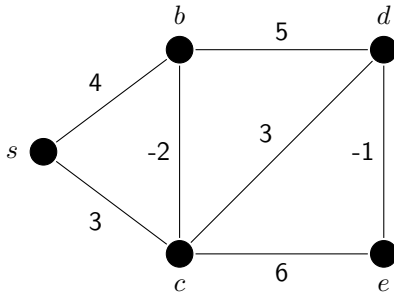
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

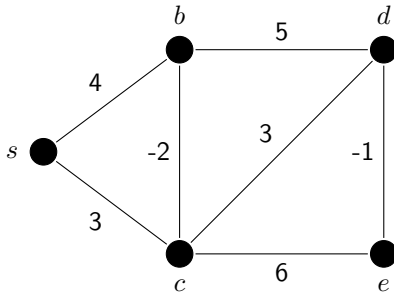
5^a : sb

6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de

2^a : bd

3^a : sc

4^a : bc

5^a : sb

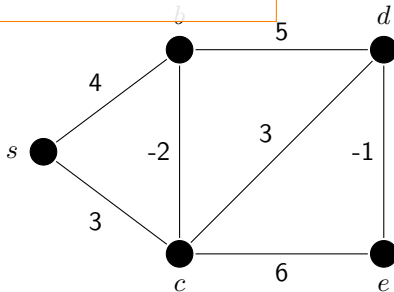
6^a : cd

7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d

ACABOU O ALGORITMO?



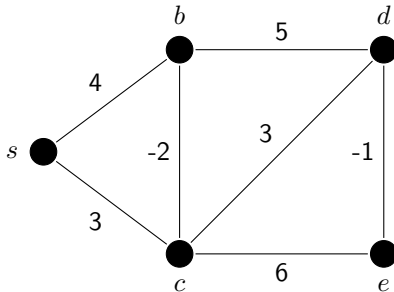
Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

- 1^a : de
2^a : bd
3^a : sc
4^a : bc
5^a : sb
6^a : cd
7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d



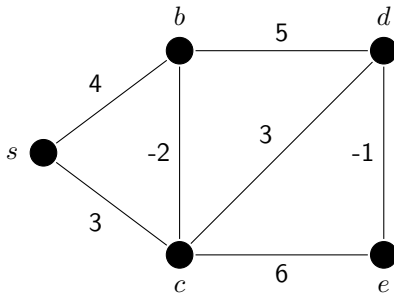
Exemplo de Execução :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : de
2^a : bd
3^a : sc
4^a : bc
5^a : sb
6^a : cd
7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d



Exemplo de Execução :: Algoritmo de Bellman-Ford

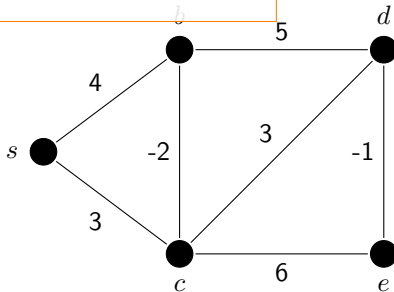
Ordem das Arestas:

- 1^a : de
2^a : bd
3^a : sc
4^a : bc
5^a : sb
6^a : cd
7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d

ACABOU O ALGORITMO?



Exemplo de Execução :: Algoritmo de Bellman-Ford

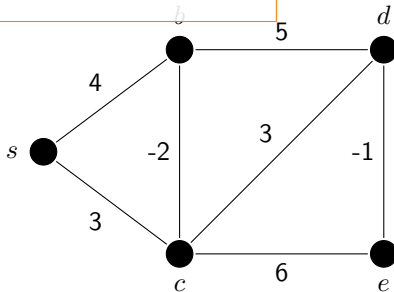
Ordem das Arestas:

- 1^a : de
2^a : bd
3^a : sc
4^a : bc
5^a : sb
6^a : cd
7^a : ce

	s	b	c	d	e
d	0	4	2	5	5

	s	b	c	d	e
π	\perp	s	b	c	d

ACABOU O ALGORITMO?



Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

	a	b	c
d	0	∞	∞

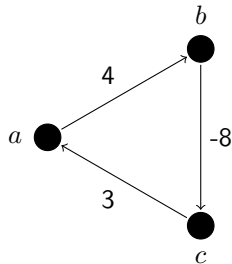
	a	b	c
π	\perp	\perp	\perp

Ordem das Arestas:

$1^a : bc$

$2^a : ca$

$3^a : ab$



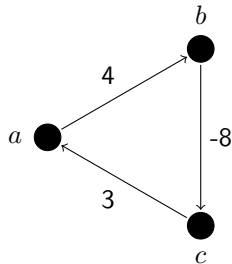
Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : bc
↓
2^a : ca
3^a : ab

	a	b	c
d	0	4	∞

	a	b	c
π	\perp	s	\perp



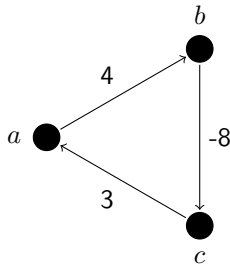
Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : bc
↓
2^a : ca
3^a : ab

	a	b	c
d	-1	4	-4

	a	b	c
π	c	s	b



Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

	a	b	c
d	-1	4	-4

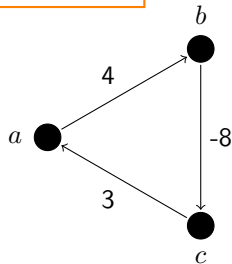
	a	b	c
π	c	s	b

Ordem das Arestas:

1^a : ACABOU O ALGORITMO?

2^a : ca

3^a : ab



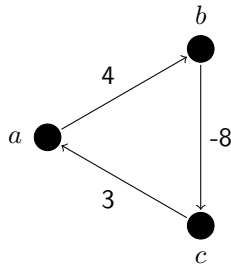
Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

Ordem das Arestas:

↓
1^a : bc
2^a : ca
3^a : ab

	a	b	c
d	-1	4	-4

	a	b	c
π	c	s	b



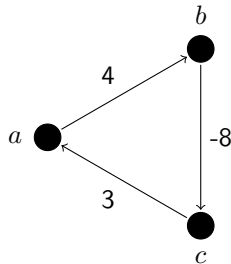
Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : bc
↓
2^a : ca
3^a : ab

	a	b	c
d	-1	4	-4

	a	b	c
π	c	s	b



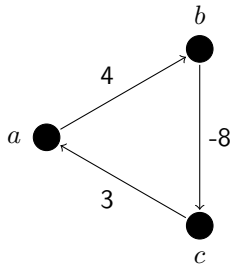
Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : bc
↓
2^a : ca
3^a : ab

	a	b	c
d	-1	4	-4

	a	b	c
π	c	s	b



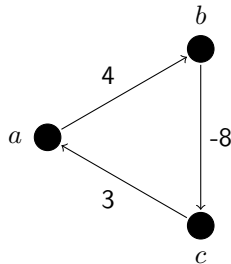
Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

Ordem das Arestas:

1^a : bc
↓
2^a : ca
3^a : ab

	a	b	c
d	-1	4	-4

	a	b	c
π	c	s	b



Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

	a	b	c
d	-1	4	-4

	a	b	c
π	c	s	b

Ordem das Arestas:

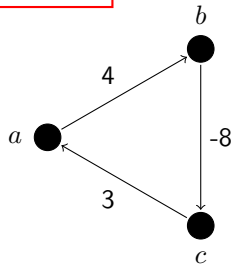


1ª : bc

2ª : ca

3ª : ab

$$d[b] > d[a] + w(a, b)!$$



Exemplo de Execução (Ruim) :: Algoritmo de Bellman-Ford

	a	b	c
d	-1	4	-4

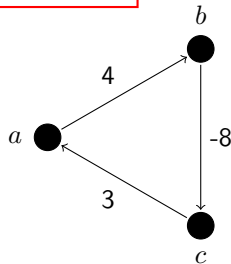
	a	b	c
π	c	s	b

Ordem das Arestas:



1^a : ab
2^a : ca
3^a : bc

Algoritmo devolve FALSE!!



Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $O(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $O(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $O(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Algoritmos para resolver CAMINHO MÍNIMO

◇ Algoritmo de Dijkstra

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(n \log n + m)$

◇ Algoritmo de Bellman-Ford

- ▷ *Single-source*
- ▷ Complexidade: $\mathcal{O}(nm)$

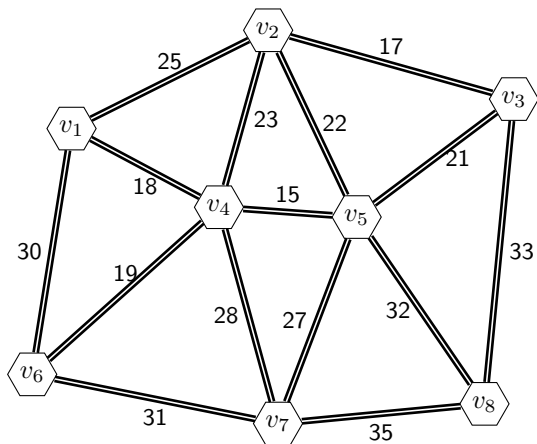
◇ Algoritmo de Floyd-Warshall

- ▷ *All-pairs*
- ▷ Complexidade: $\mathcal{O}(n^3)$

Dúvidas?

Aula que vem...

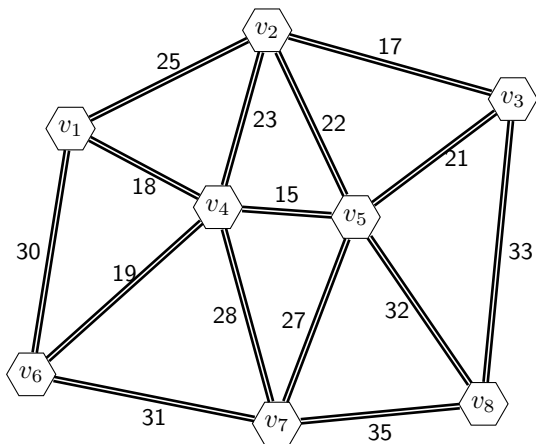
Árvore Geradora Mínima



ÁRVORE GERADORA MÍNIMA

Entrada: Um grafo ponderado $G = (V, U, w)$.

Pergunta: Qual é a árvore $T \subseteq G$ de custo mínimo que gera G ?



ÁRVORE GERADORA MÍNIMA

Entrada: Um grafo ponderado $G = (V, U, w)$.

Pergunta: Qual é a árvore $T \subseteq G$ de custo mínimo que gera G ?