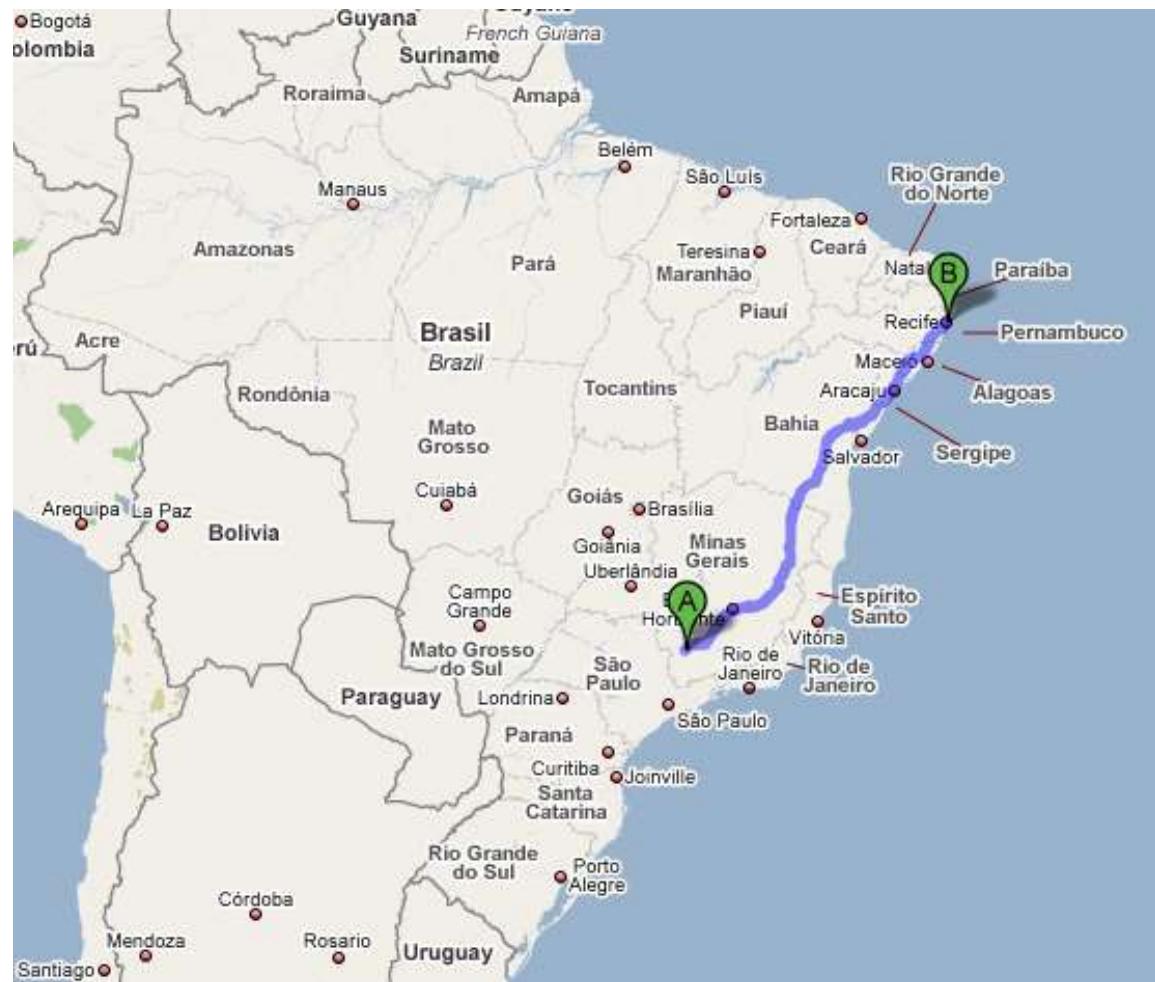


Caminho Mínimo

- Suponha que você deseja encontrar um caminho mais curto de Alfenas/MG para Recife/PE;



Caminhos mais curtos de origem única

- A princípio, podemos imaginar um algoritmo que enumera todas as possíveis rotas;
 - Se você conhece TODAS as possibilidades a tarefa é apenas computar a distância de cada uma delas e **escolher aquela que oferece o menor trajeto.**
 - O problema é computável;
 - Pode ser resolvido por uma máquina de Turing ou máquina equivalente.
 - Obviamente, tomando o cuidado de **não considerar trajetos com ciclos**; Seria equivalente a processar um autômato com ciclo(s). A linguagem correspondente possui infinitas palavras (rotas) – apesar das palavras possuírem tamanho finito.

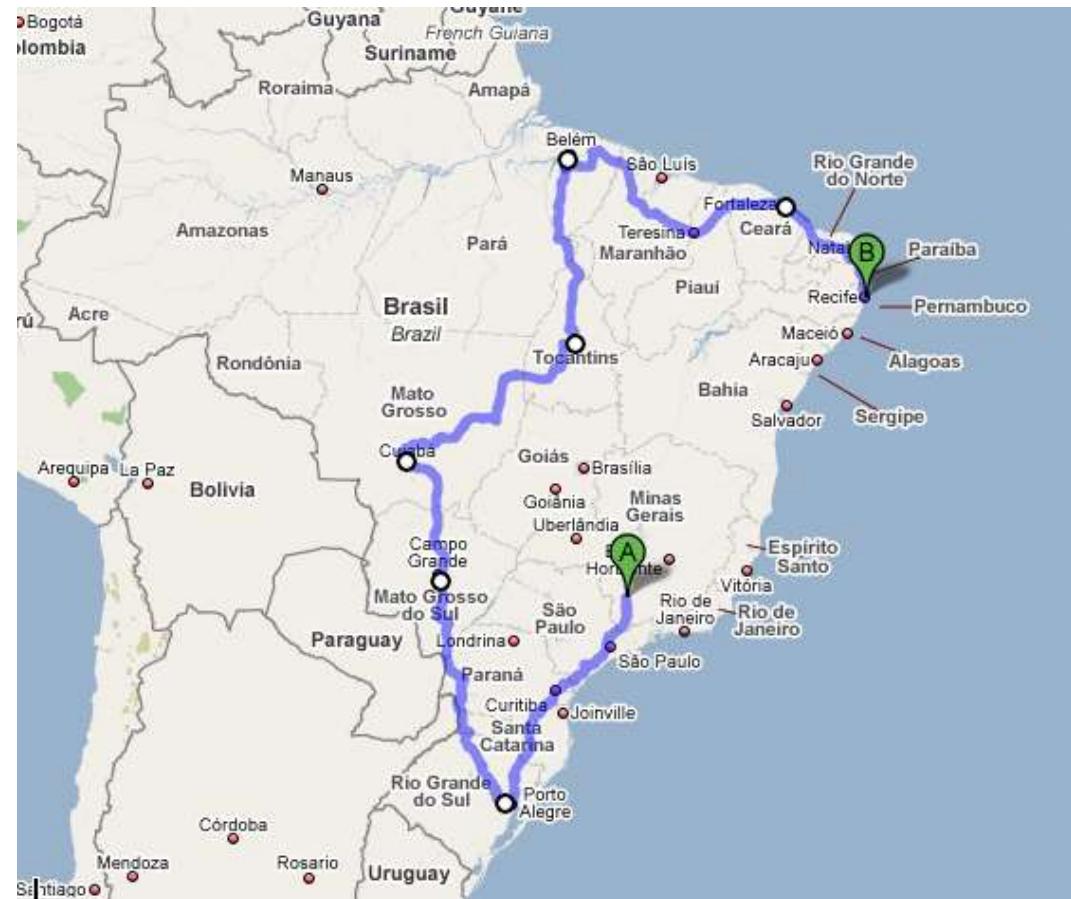
Caminhos mais curtos de origem única

- O problema da abordagem da **enumeração**, é que haverá **milhões/bilhões de possibilidades**;
- Sendo que a maioria não vale a pena considerar...
- Veja exemplo na figura...



Caminhos mais curtos de origem única

- Podemos realizar obviamente uma enumeração implícita.
- **Se a minha rota incompleta atual já possui distância maior que a minha melhor rota completa até então conhecida, então, não precisa continuar com o caminho incompleto atual.**
 - Política de corte em uma busca em profundidade.



Caminhos mais curtos de origem única

- Para o problema dos caminhos mais curtos, temos um grafo orientado ponderado

$$G = (V, A)$$

- Com função de peso

$$w: A \rightarrow \mathbb{R}$$

- Mapeando as arestas em valores reais.

Caminhos mais curtos de origem única

- O peso do caminho p

$$p = \langle v_0, v_1, v_2, \dots, v_k \rangle$$

- é o somatório dos pesos de suas arestas

$$w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$$

Caminhos mais curtos de origem única

- Definimos peso do caminho mais curto de u até v como:

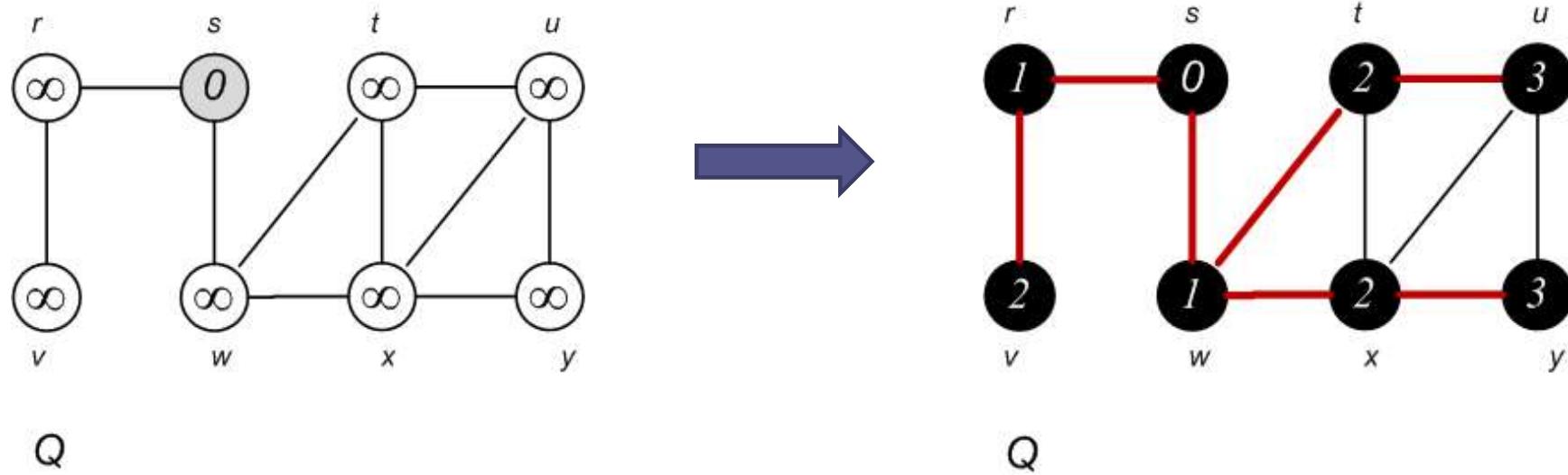
$$\delta(u, v) = \begin{cases} \min\{w(p) : u \xrightarrow{p} v\}, & \text{se existe caminho de } u \text{ até } v \\ \infty, & \text{caso contrário} \end{cases}$$

- Assim, um **caminho mais curto** de u para v é definido como qualquer caminho p onde

$$w(p) = \delta(u, v)$$

Caminhos mais curtos de origem única

- Relembrando...
 - O algoritmo de busca em largura (BFS) é um algoritmo que calcula caminhos mais curtos em grafos não ponderados...



Caminhos mais curtos de origem única

- Variantes do Problema
 - Caminhos mais curtos de origem única;
 - Caminhos mais curtos de destino único;
 - Caminho mais curto de um único par;
 - Caminhos mais curtos de todos para todos;

Sub-estrutura ótima



Caminhos mais curtos de origem única

- Sub-estrutura ótima de um caminho mais curto
 - Serve de base para todos os algoritmos exatos que calculam as variantes do caminho mais curto;
 - Seja $p = \langle v_1, v_2, \dots, v_k \rangle$ um caminho mais curto de v_1 para v_k ;
 - Para quaisquer i e j tais que $1 \leq i <= j \leq k$, e seja $p_{ij} = \langle v_i, v_{i+1}, \dots, v_j \rangle$ o subcaminho p' desde o vértice v_i até v_j .
 - Então p_{ij} é um caminho mais curto de v_i para v_j .

Caminhos mais curtos de origem única

- Subestrutura ótima de um caminho mais curto

Sub-caminho de Aracaju para Maceió é ótimo;

Sub-caminho de Maceió para Recife também é ótimo.



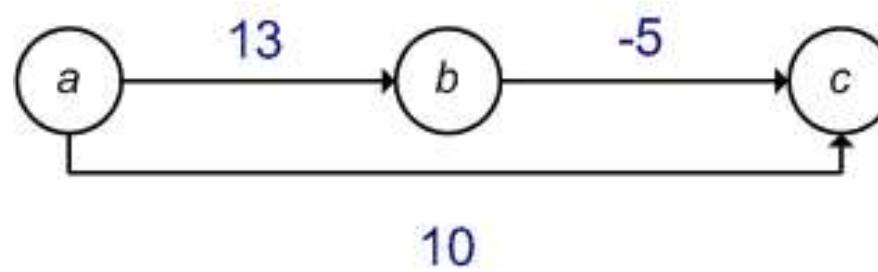
Arestas de peso negativo em um Algoritmo de Caminho Mínimo



Caminhos mais curtos de origem única

- Arestas de peso negativo

- Grafos podem ter arestas de peso negativo;
- Qual é o custo do menor caminho de a até c ?

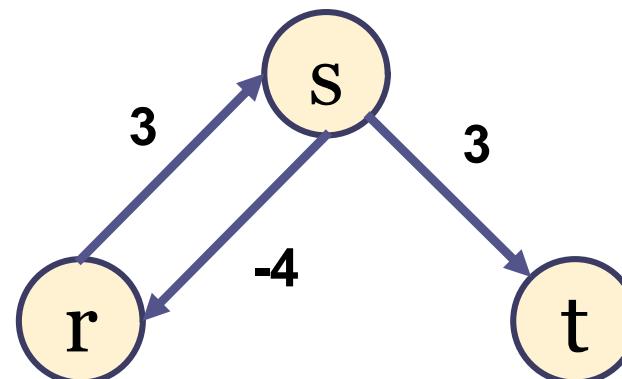


- Você visualiza alguma dificuldade envolvendo arestas de pesos negativos?

Caminhos mais curtos de origem única

- Arestas de peso negativo

- Se existe um ciclo de peso negativo acessível a partir da origem s , os pesos dos caminhos mais curtos perdem a referência;
- Sempre será possível encontrar um caminho mais curto de peso menor que o já encontrado;



Caminhos mais curtos de origem única

- Arestas de peso negativo

- Se existe um ciclo de peso negativo em algum caminho desde s até v , definimos

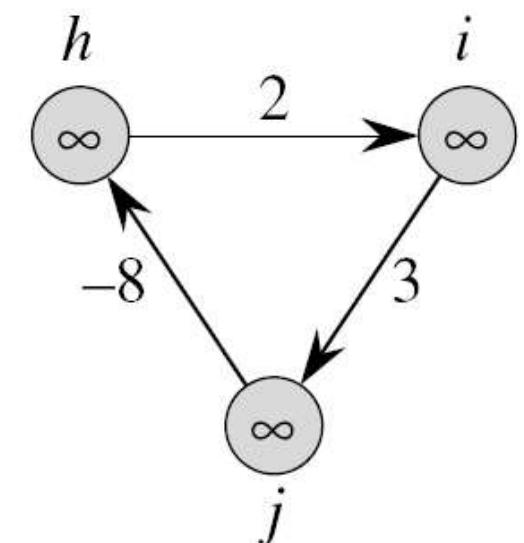
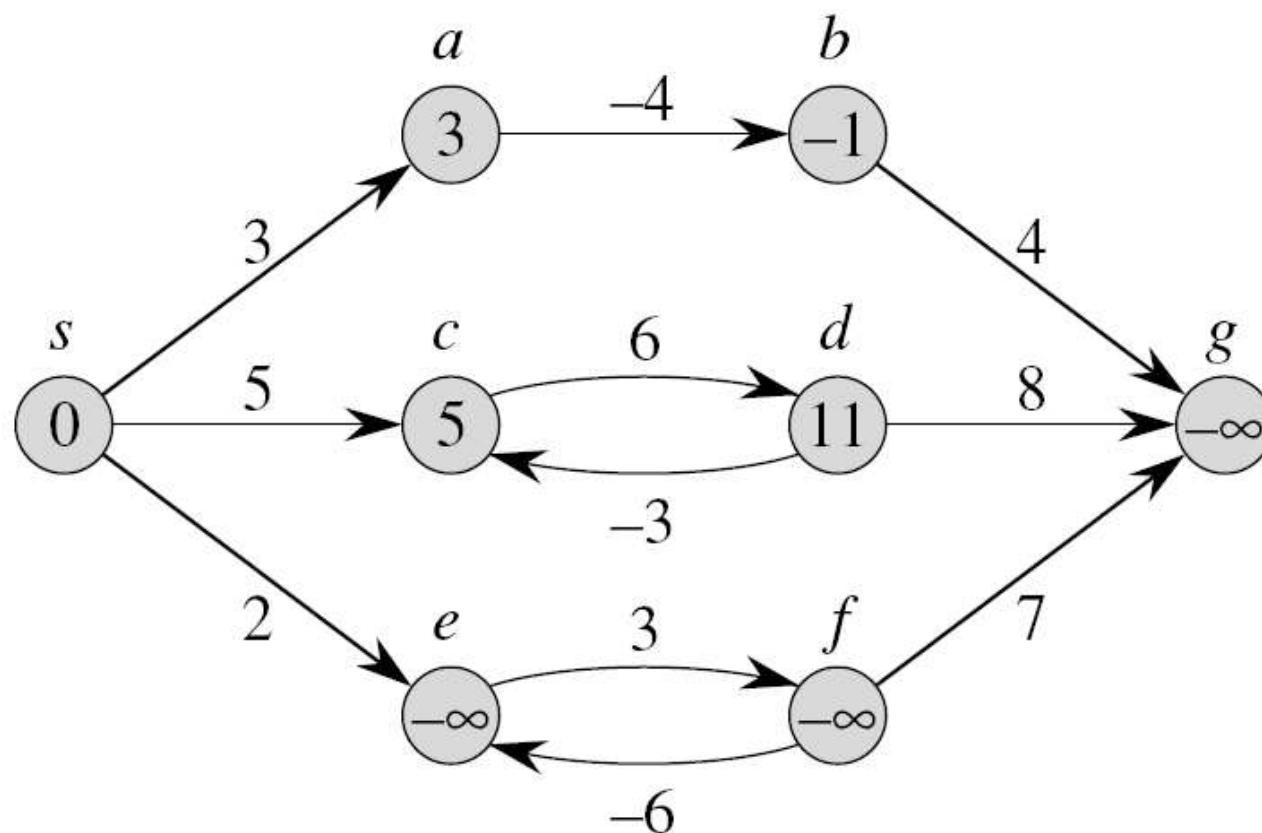
$$\delta(s, v) = -\infty$$

- Relembrando...

$$\delta(u, v) = \begin{cases} \min \{ w(p) : u \xrightarrow{p} v \}, & \text{se existe caminho de } u \text{ até } v \\ \infty, & \text{caso contrário} \end{cases}$$

Caminhos mais curtos de origem única

- Arestas de peso negativo



Caminhos mais curtos de origem única

- Arestas de peso negativo
 - O algoritmo de Dijkstra assume que todos os pesos de arestas no grafo de entrada são não negativos;
 - Ideal para aplicação em mapas rodoviários;
 - Algoritmo mais aplicado na prática em sistemas comerciais;
 - O algoritmo de Bellman-Ford permite a existência de arestas de peso negativo, e produz a resposta correta;
 - *Não entra em loop infinito.*

Caminhos mais curtos de origem única

- Ciclos
 - Como vimos, caminhos mais curtos não podem conter ciclos de peso negativo;
 - Mas eles podem conter ciclo de peso positivo?

Caminhos mais curtos de origem única

- Representação de caminhos mais curtos

- Assim como na busca em largura (BFS) iremos utilizar os vetores d e π para recuperar caminhos após a aplicação dos algoritmos;
- Relembrando:

$$\pi[u] = \begin{cases} \text{pai do vértice } u, \text{ se } u \text{ é alcançável;} \\ \text{NULL, em caso contrário.} \end{cases}$$

$$d[u] = \begin{cases} \delta(s, u), \text{ se } u \text{ é alcançável;} \\ \infty, \text{ em caso contrário.} \end{cases}$$

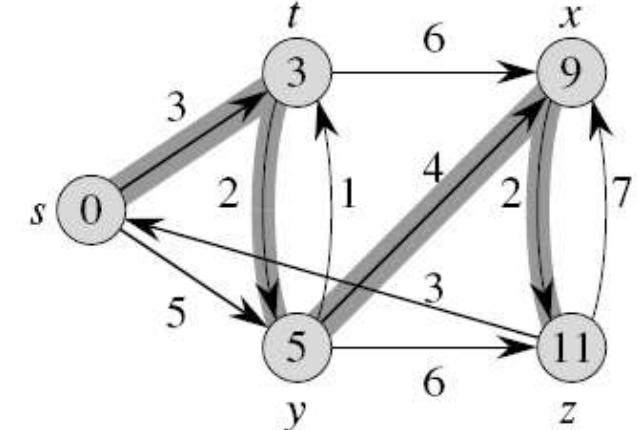
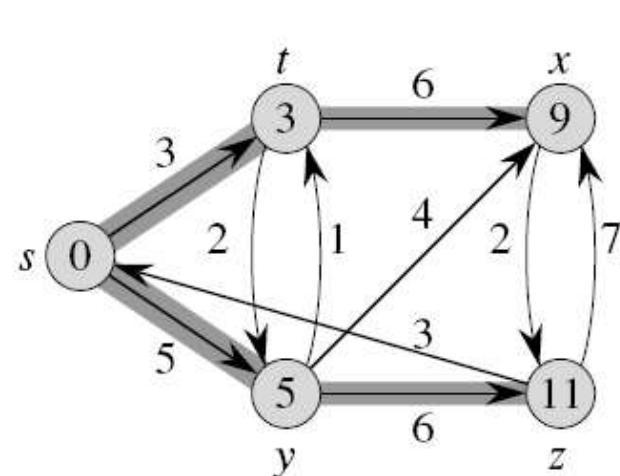
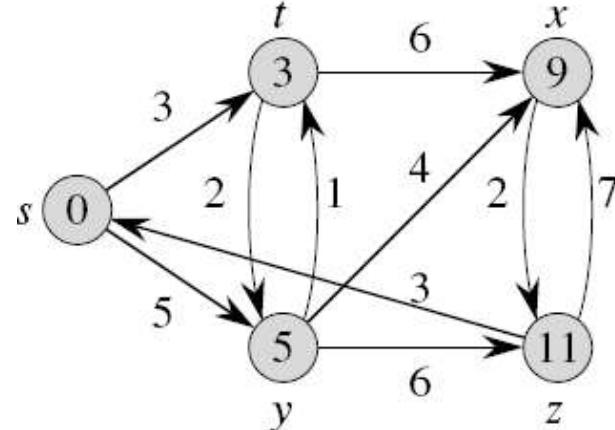
Caminhos mais curtos de origem única

- Árvore de caminhos mais curtos

$$G' = (V', A')$$

$$V' \subseteq V \text{ e } A' \subseteq A$$

- V' é o conjunto de vértices acessíveis a partir de s no grafo;
- G' forma uma árvore enraizada com raiz s ;
- Para todo vértice do grafo, o único caminho simples desde s até v em G' é um caminho mais curto desde s até v em G .



Caminhos mais curtos de origem única

- Inicialização dos vetores

- Os algoritmos que veremos para esta classe de problemas usam um método para inicializar os vetores auxiliares:

INICIALIZA($G = (V, A)$, s)

para cada $v \in V$

$d[v] = \infty$

$\pi[v] = \text{NULL}$

fim para

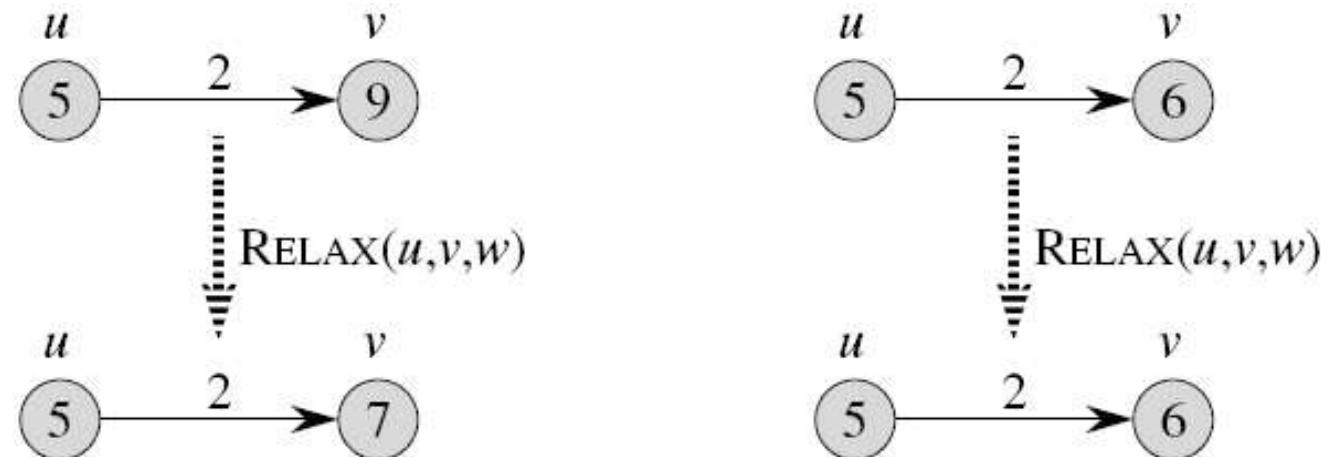
$d[s] = 0$

fim

Caminhos mais curtos de origem única

- Técnica do relaxamento

- “O processo de relaxar uma aresta (u,v) consiste em testar se podemos melhorar o caminho mais curto para v encontrado até agora pela passagem de u e, neste caso, atualizar $d[v]$ e $\pi[v]$.”



Caminhos mais curtos de origem única

- Técnica do relaxamento

$\text{RELAXA}(u, v, w)$

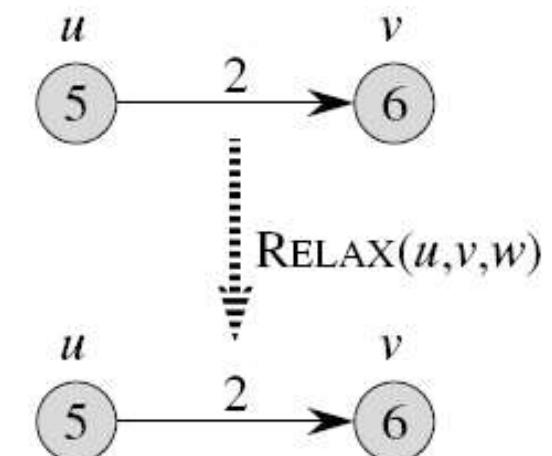
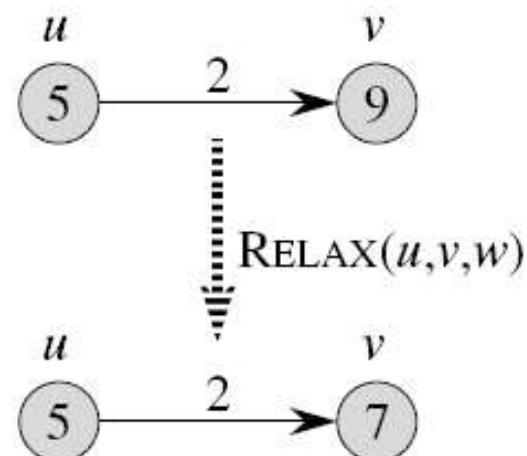
se $d[v] > (d[u] + w(u, v))$ então

$d[v] \leftarrow d[u] + w(u, v)$

$\pi[v] = u$

fim se

fim



Algoritmo de Dijkstra

- Djikstra:

- Resolve o problema do caminho mais curto de origem única em:
 - um **grafo direcionado, ponderado** $G=(V,A)$;
 - Sendo que todos os **pesos** das arestas são **não negativos**;
 - O grafo **pode conter ciclos**.

Algoritmo de Dijkstra

- É o algoritmo de caminho mais curto mais aplicado para **problemas rodoviários** porque
 - aceita **ciclos** (comum nas estradas) e
 - **não possui arestas de peso negativo** (não existe distância negativa entre dois pontos quaisquer em um mapa).
- *Como não possui arestas negativas, consequentemente não possui ciclos de peso negativo.*

Algoritmo de Dijkstra

- Comentário:
- O algoritmo de Dijkstra é um algoritmo **GULOSO**;
- Assim como outros métodos vistos na disciplina, apesar de ser guloso, ele é **exato**:
 - encontra sempre o menor caminho da raiz para todos os nós do grafo.

Algoritmo de Dijkstra

- O algoritmo basicamente trabalha com dois tipos de vértices.
 - Aqueles que já é conhecida a menor distância para a raiz, e
 - Aqueles que a distância conhecida ainda é provisória (sem garantia de otimalidade).

Algoritmo de Dijkstra

- Estruturas utilizadas:

$\pi[u] \rightarrow$ pai do vértice u

$d[u] \rightarrow$ Distância da origem até u

$Q \rightarrow$ nós com distância provisória

$S \rightarrow$ nós com distância definitiva

Algoritmo de Dijkstra

- Relembrando dois métodos básicos utilizados:

INICIALIZA($G = (V, A), s$)

 para cada $v \in V$

$d[v] = \infty$

$\pi[v] = \text{NULL}$

 fim para

$d[s] = 0$

fim

RELAXA(u, v, w)

 se $d[v] > (d[u] + w(u, v))$ então

$d[v] \leftarrow d[u] + w(u, v)$

$\pi[v] = u$

 fim se

fim

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A)$, w , s)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow \text{extrair M inino}(Q)$

$S \leftarrow S \cup \{u\}$

para cada $v \in Adj[u]$

relaxa(u, v, w)

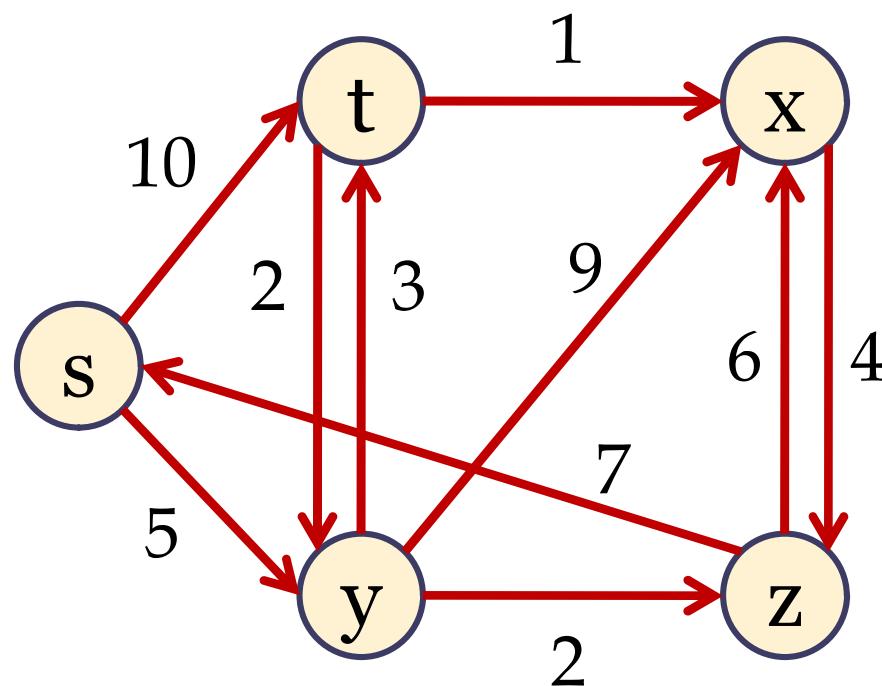
fim para

fim enquanto

fim

Algoritmo de Dijkstra

- Considere o grafo:



Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

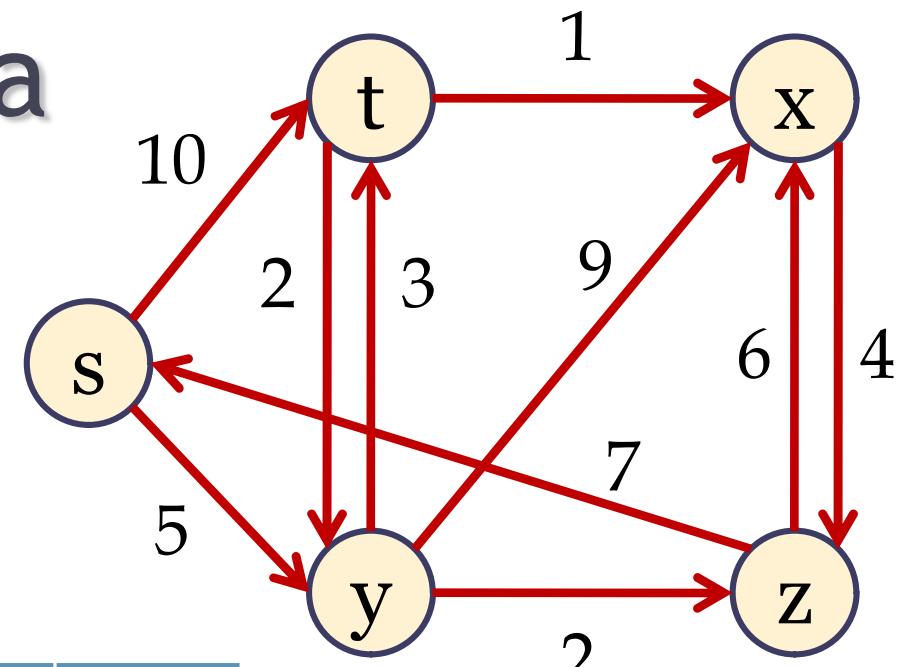
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d					
π					
Q					
S					

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

→ INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

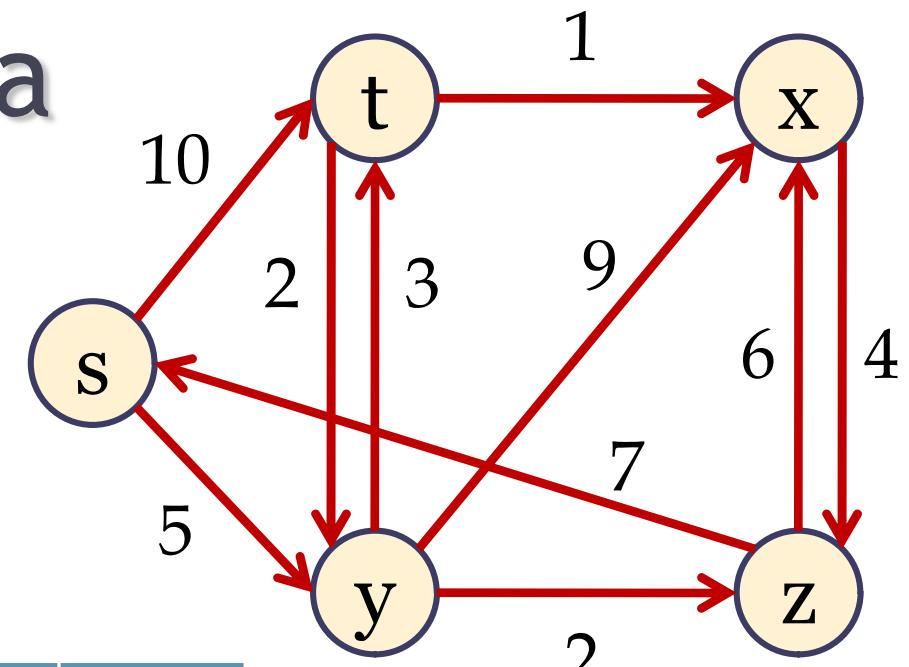
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d					
π					
Q					
S					

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

→ INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

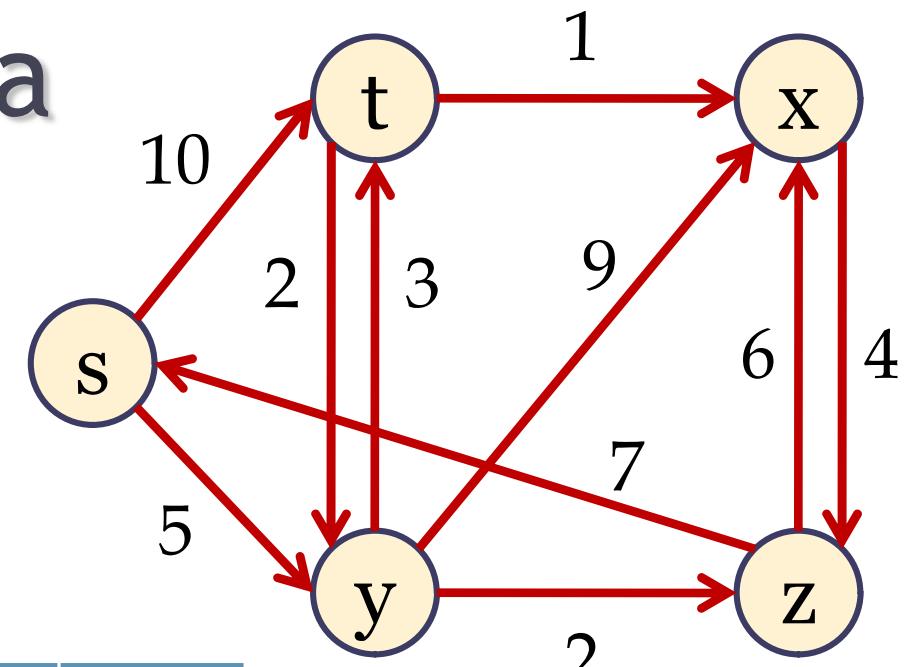
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q					
S					

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

 INICIALIZA(G, s)

$\rightarrow S \leftarrow \{ \}$

$Q \leftarrow V$

 Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

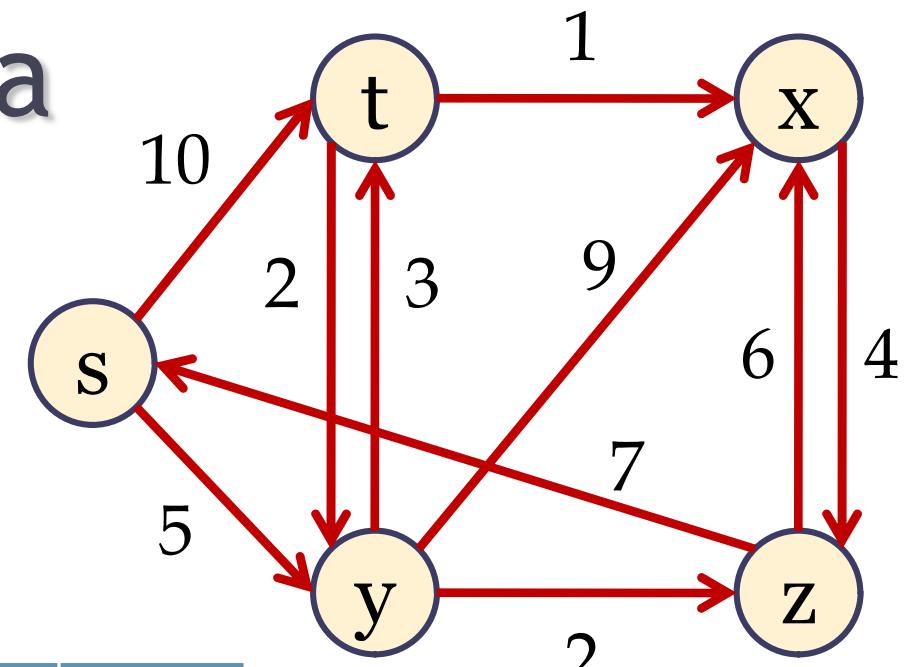
 para cada $v \in Adj[u]$

 relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q					
S					

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

 INICIALIZA(G, s)

$\rightarrow S \leftarrow \{ \}$

$Q \leftarrow V$

 Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

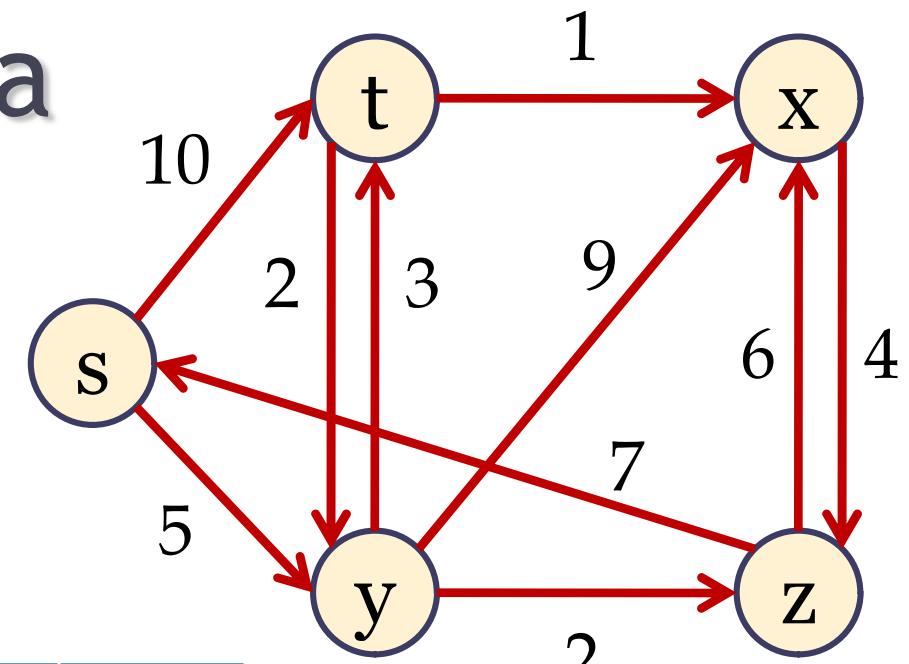
 para cada $v \in Adj[u]$

 relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q					
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$\rightarrow Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

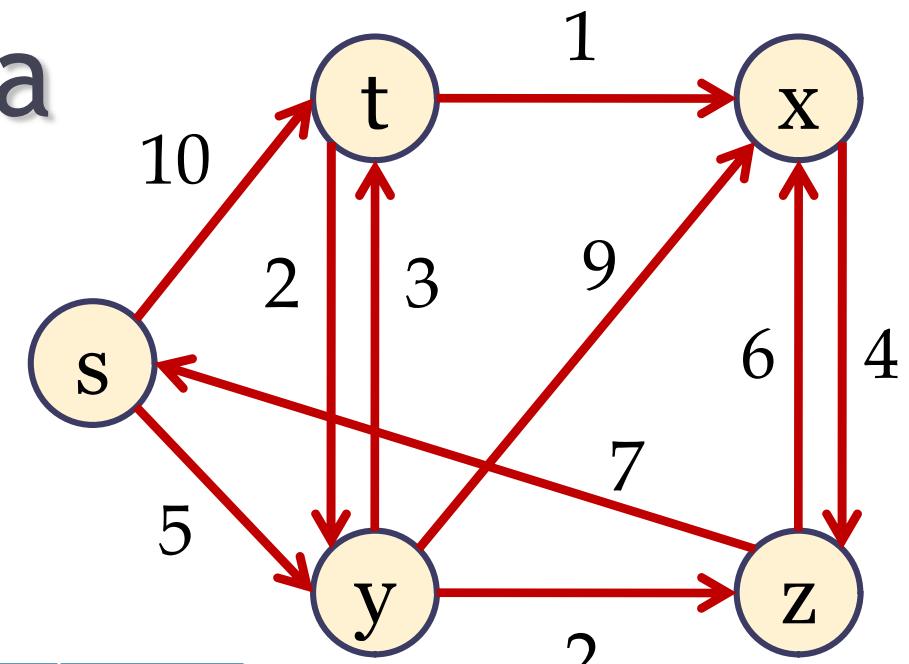
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q					
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$\rightarrow Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

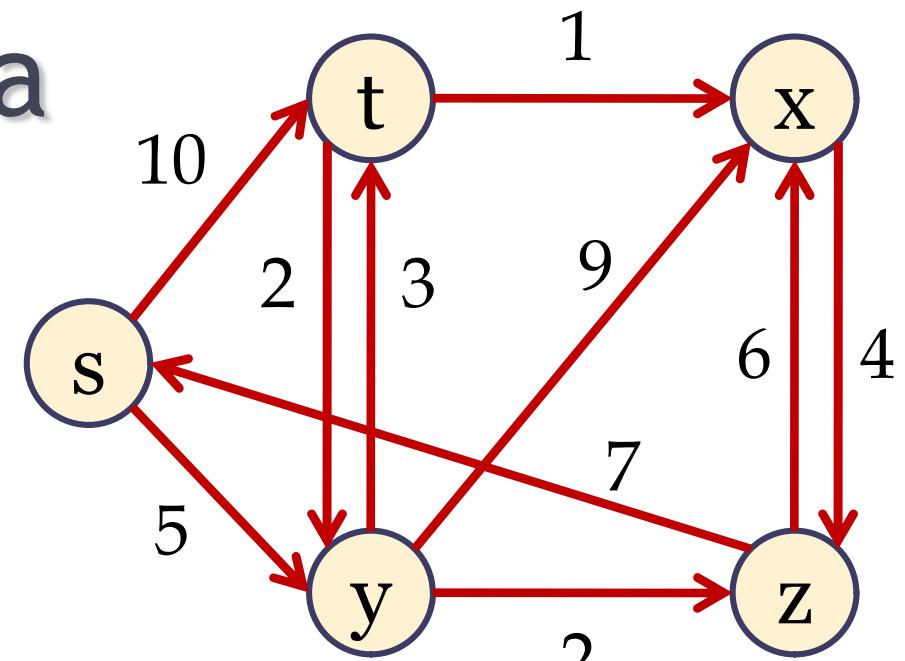
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	X	X	X	X	X
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

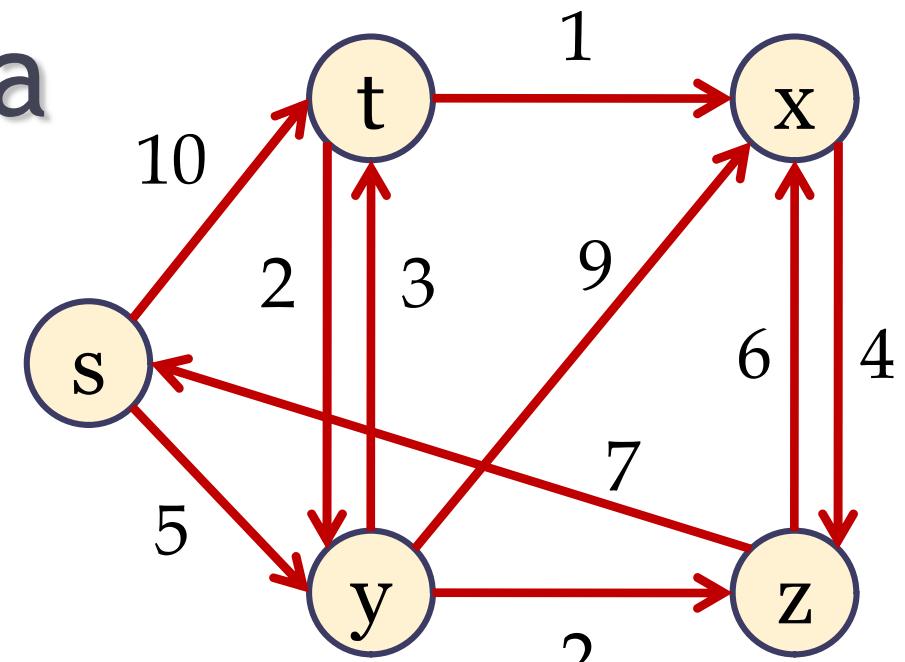
para cada $v \in \text{Adj}[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	X	X	X	X	X
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

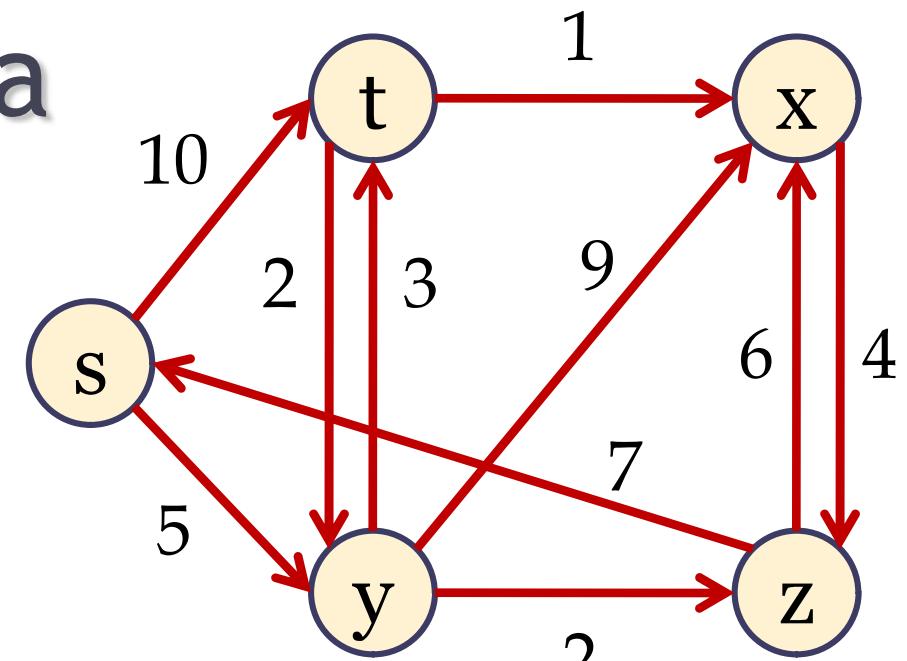
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	X	X	X	X	X
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

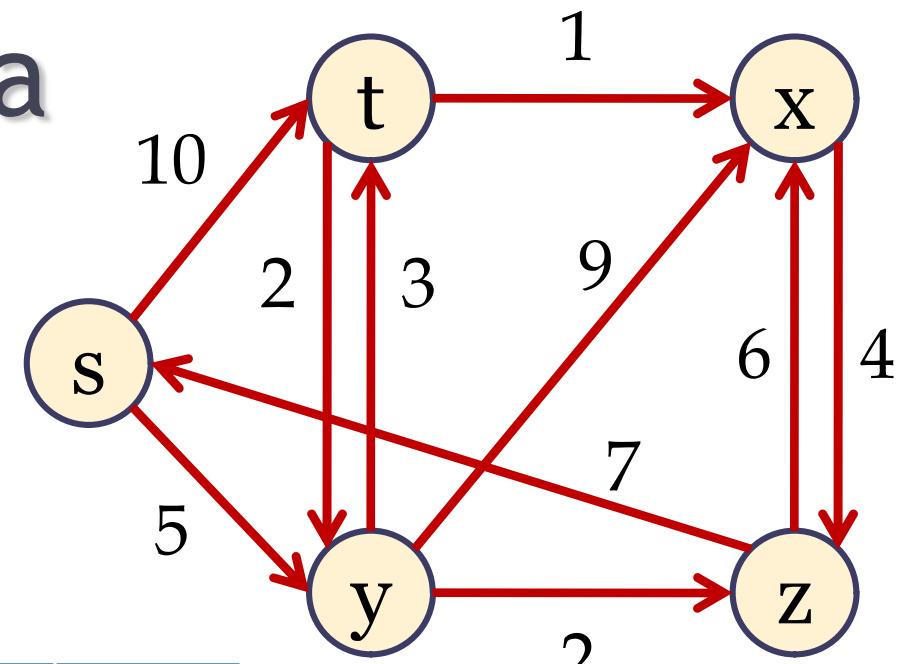
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	---	X	X	X	X
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

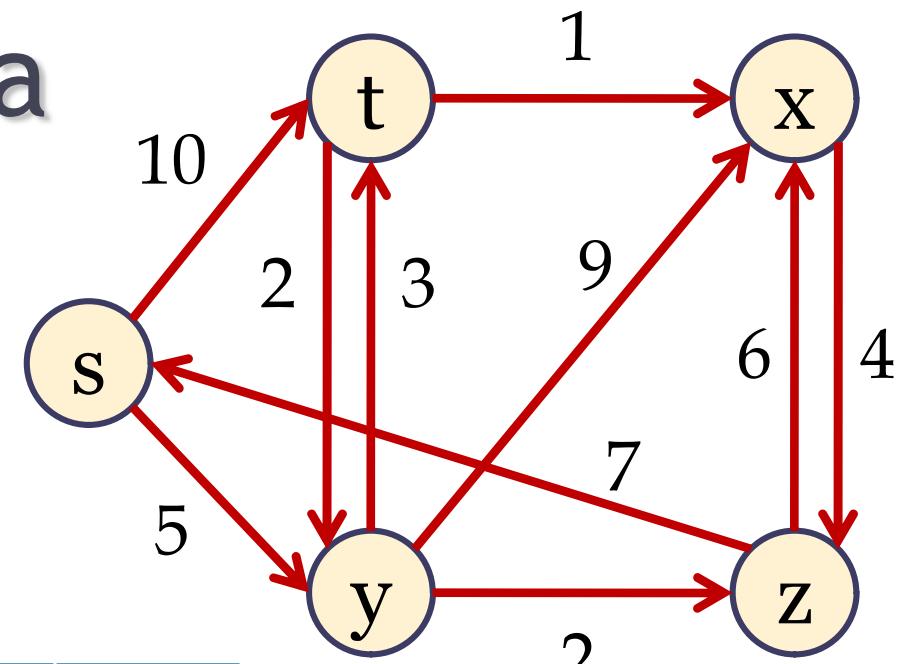
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	---	X	X	X	X
S	---	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

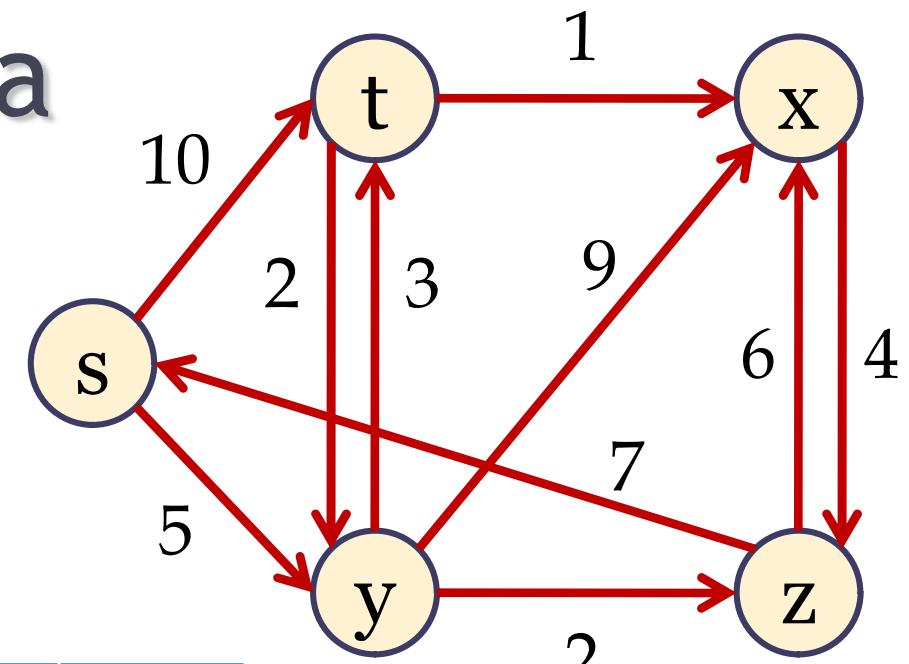
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	---	X	X	X	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

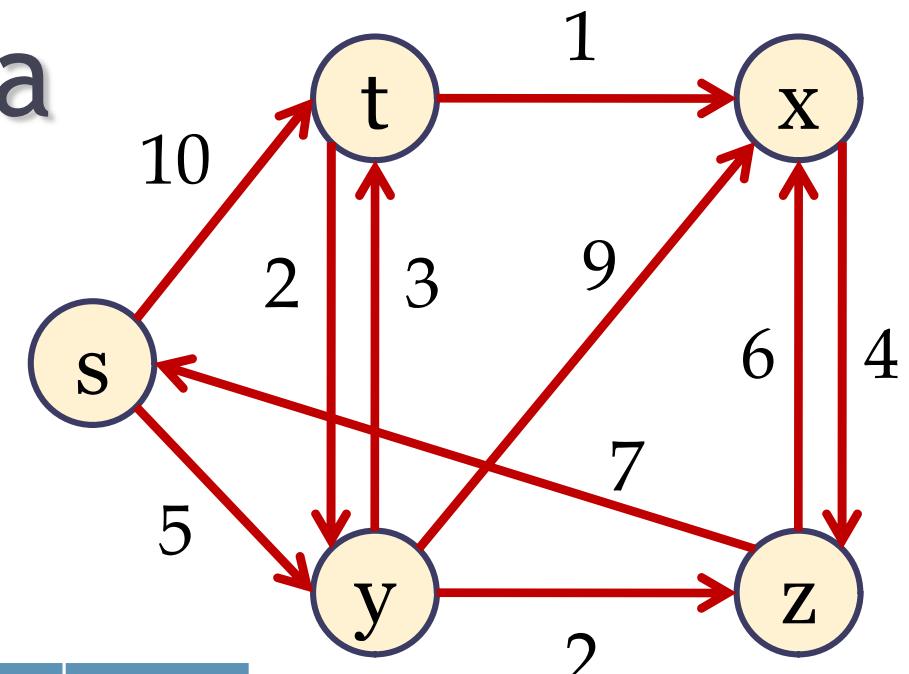
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	∞	∞	∞	∞
π	NULL	NULL	NULL	NULL	NULL
Q	---	X	X	X	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

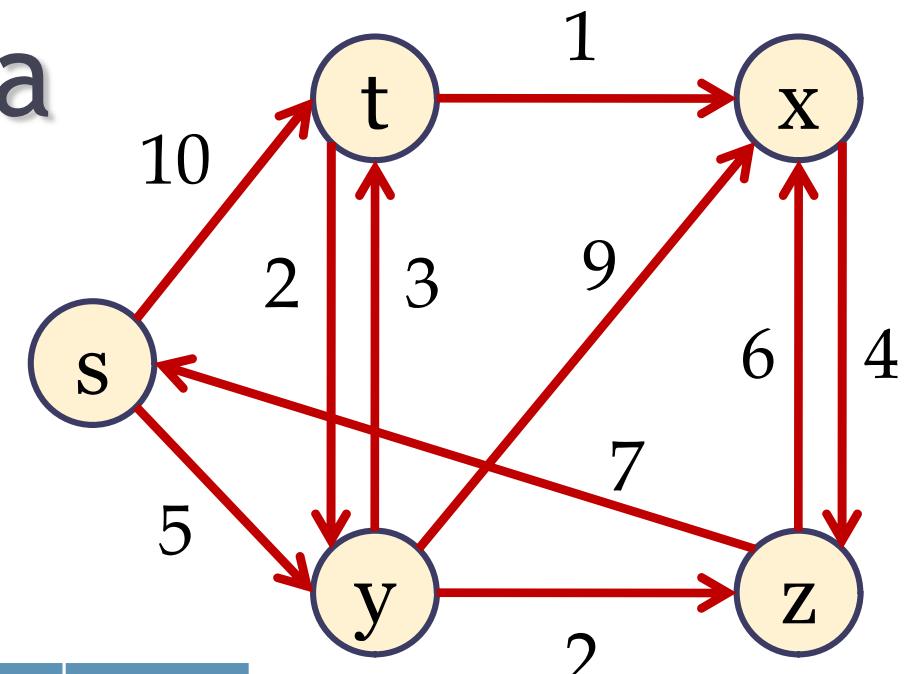
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	X	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

→ Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

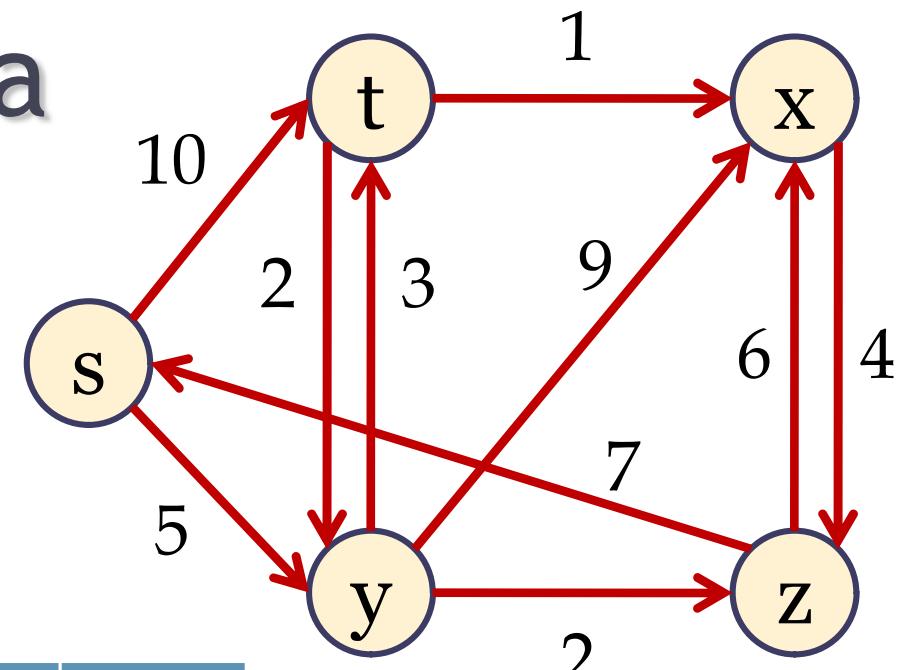
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	X	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

→ $u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

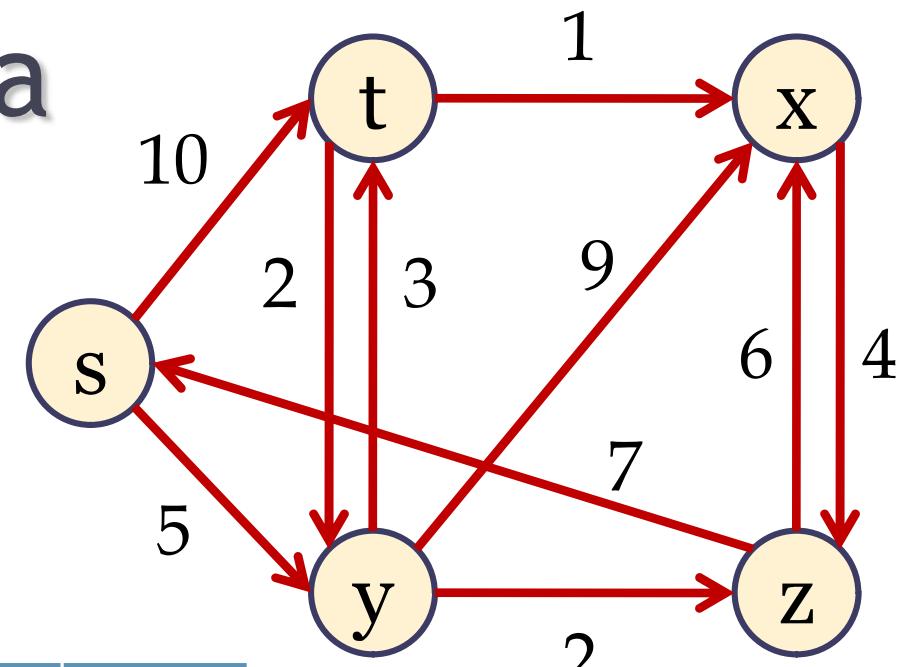
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	X	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

→ $u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

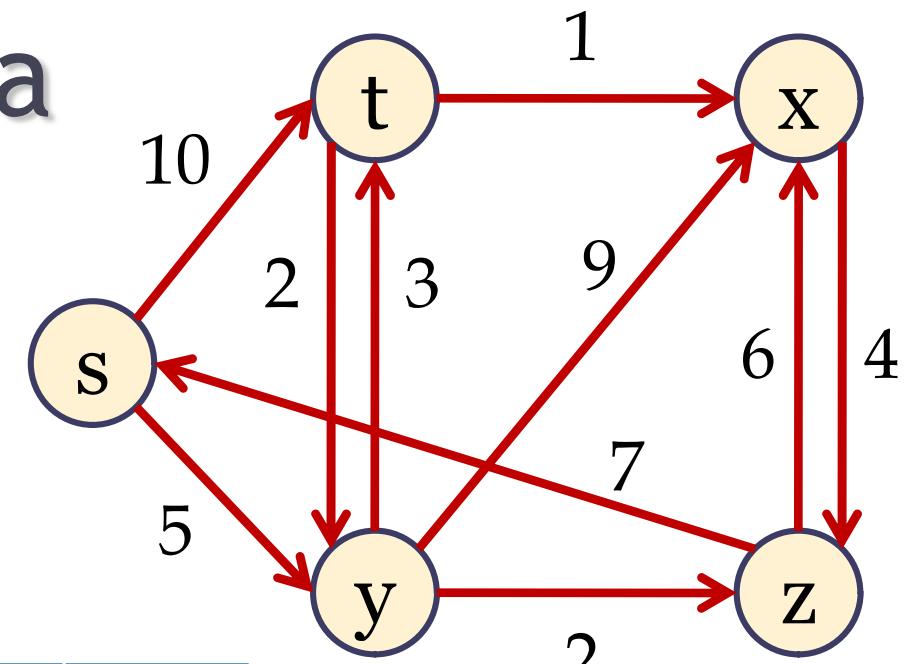
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	s

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	X	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

→ $u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

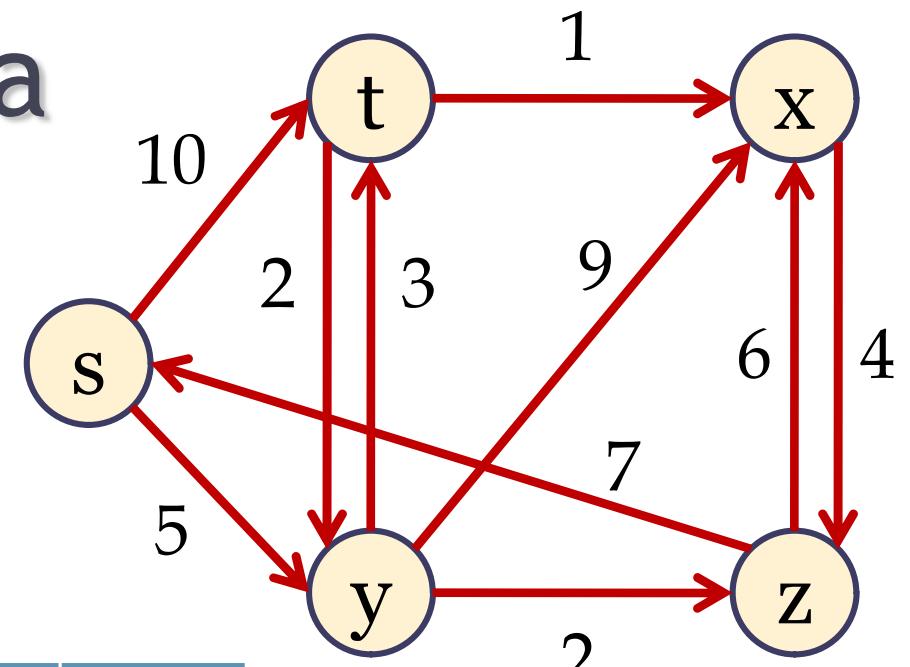
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	x	x	---	x
S	x	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

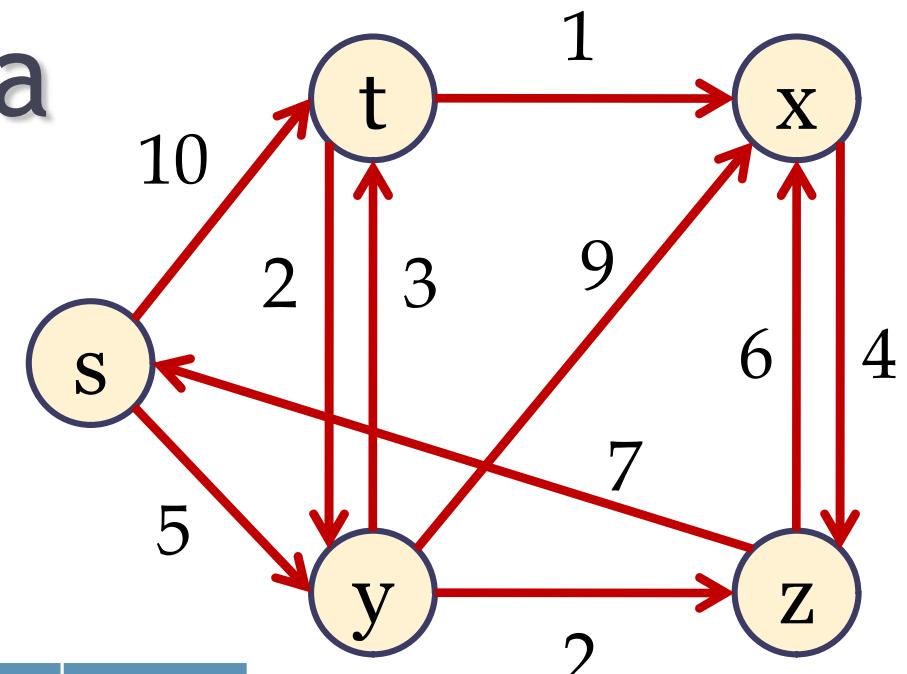
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	---	X
S	X	---	---	---	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

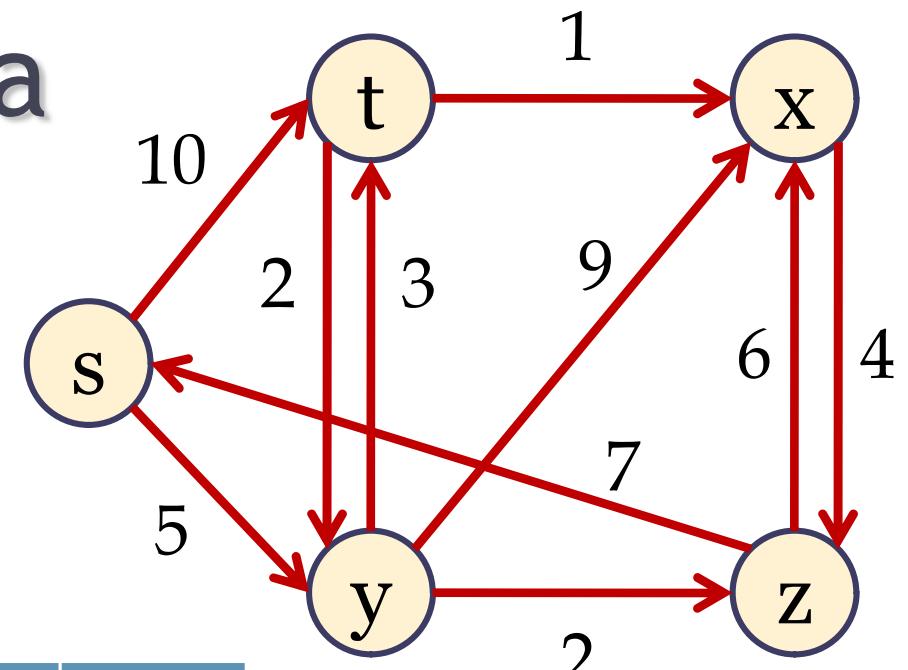
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

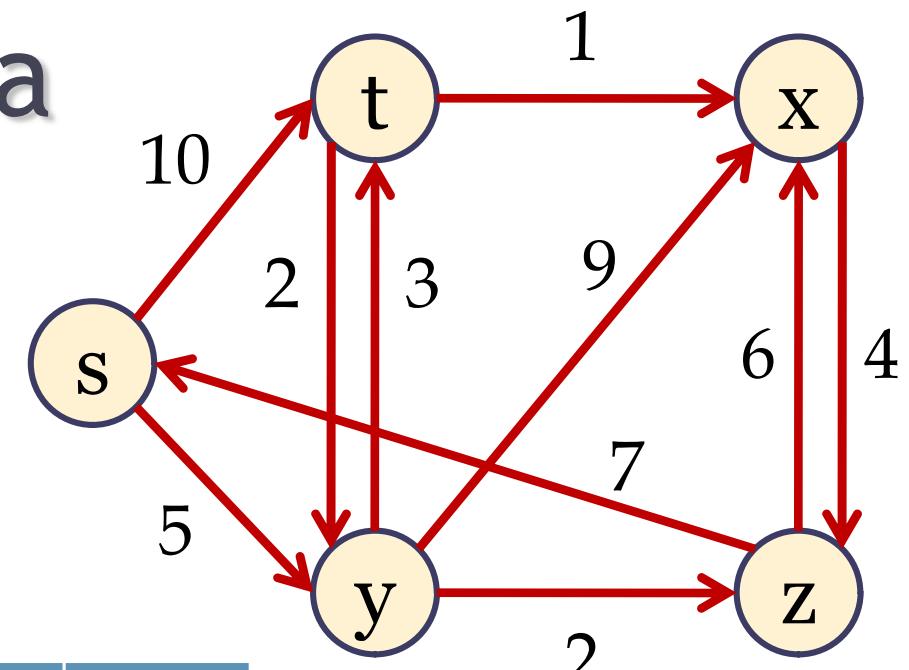
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

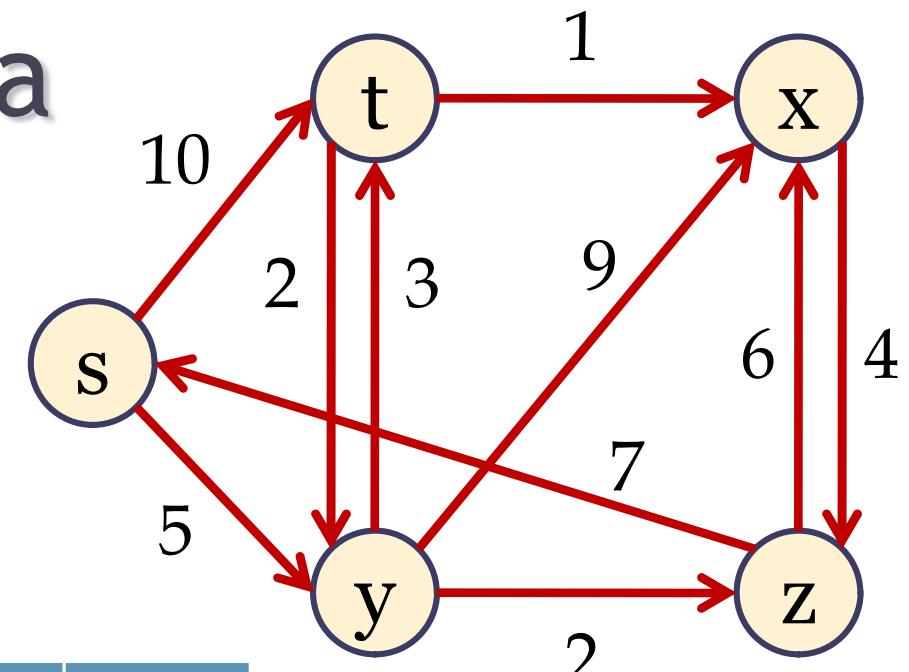
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	10	∞	5	∞
π	NULL	s	NULL	s	NULL
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

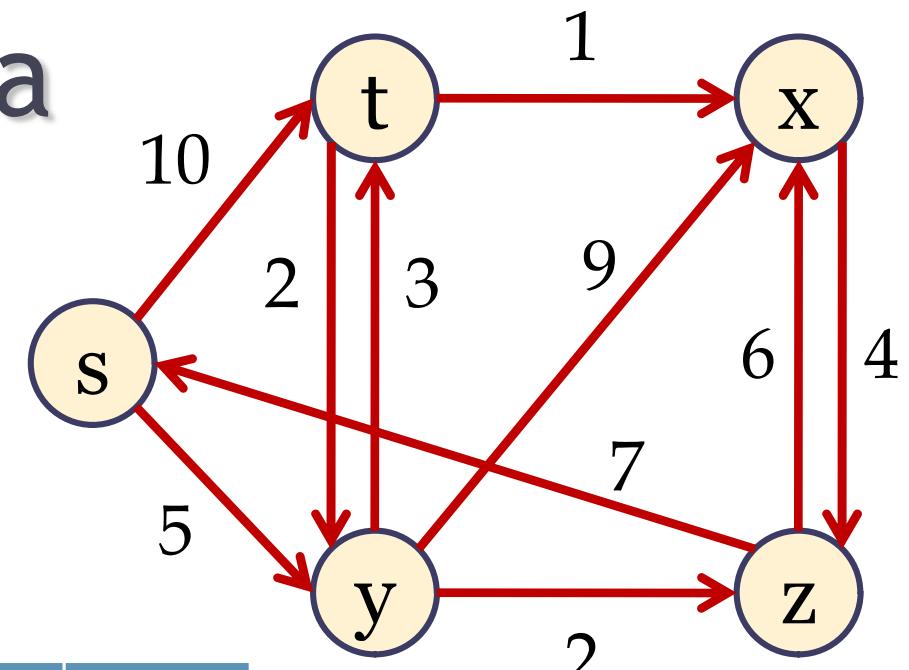
→ para cada $v \in Adj[u]$

→ relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

→ Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

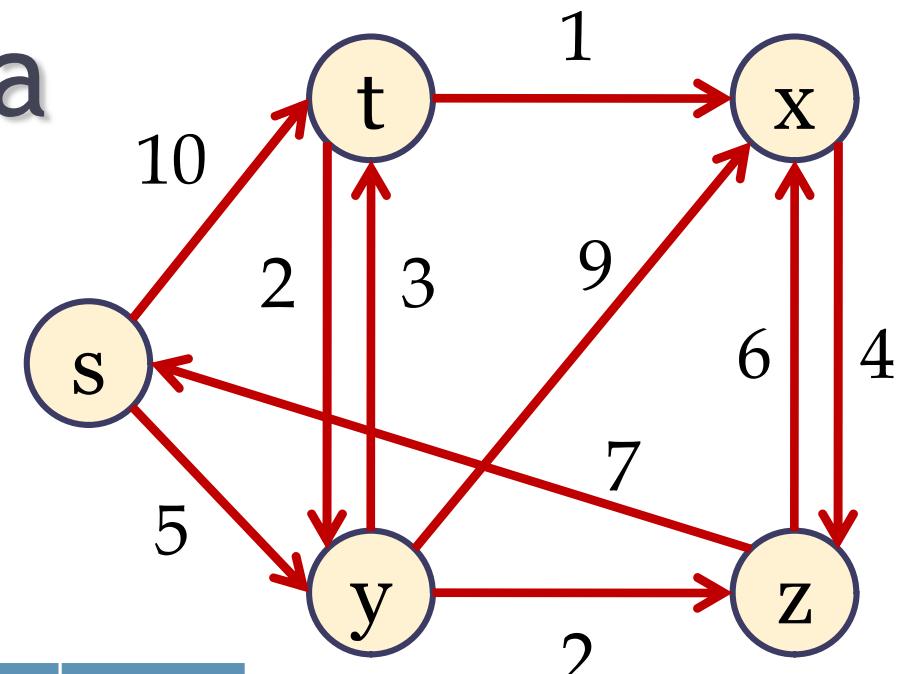
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

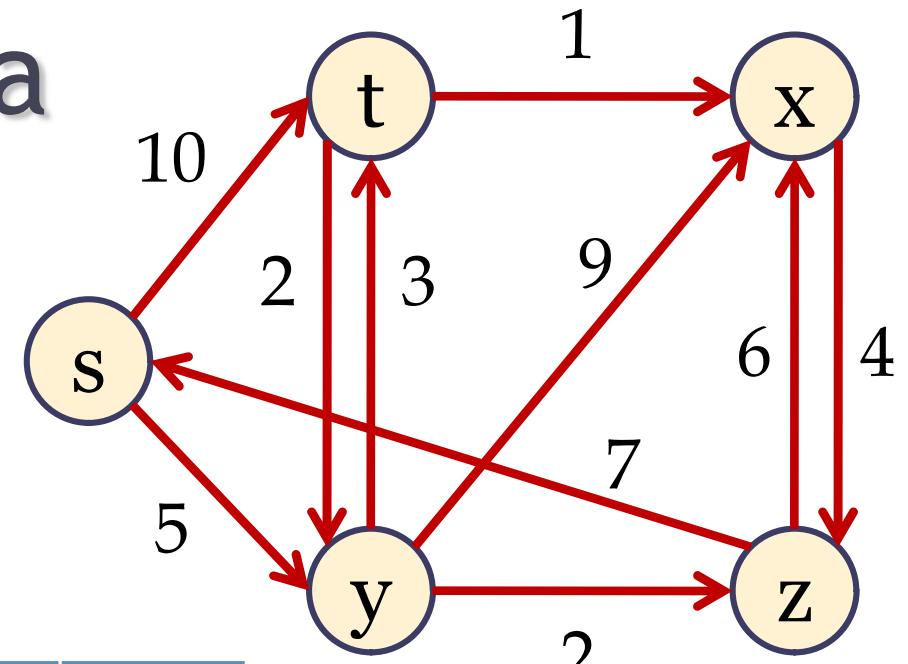
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

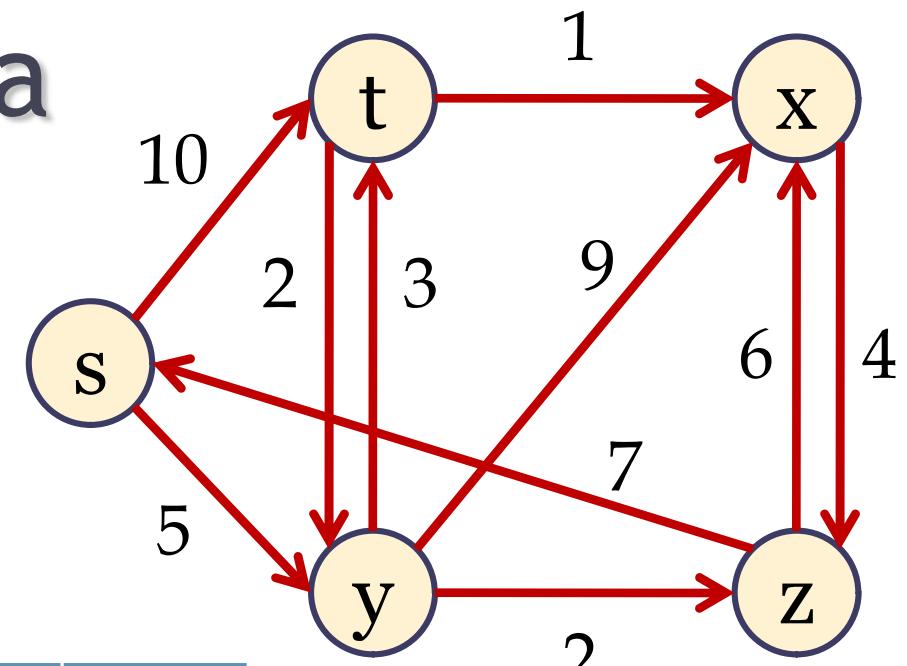
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	y

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	X
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

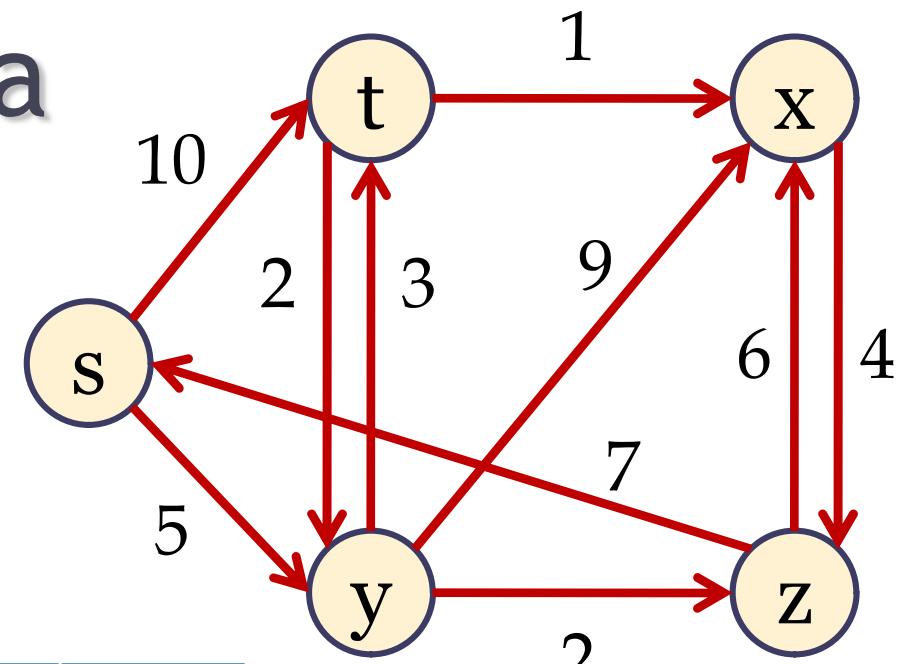
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	---
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

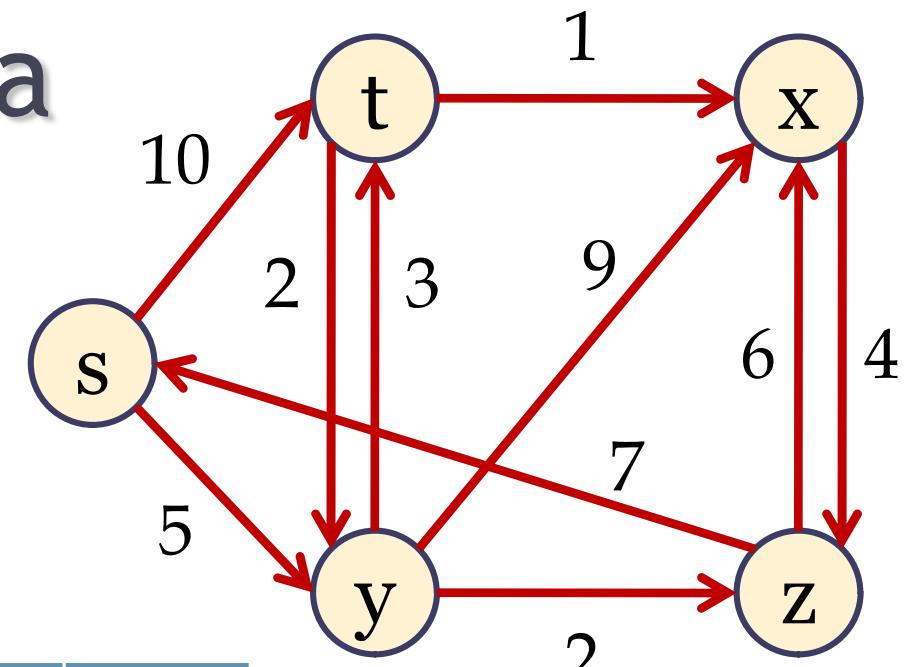
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	---
S	X	---	---	X	---

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

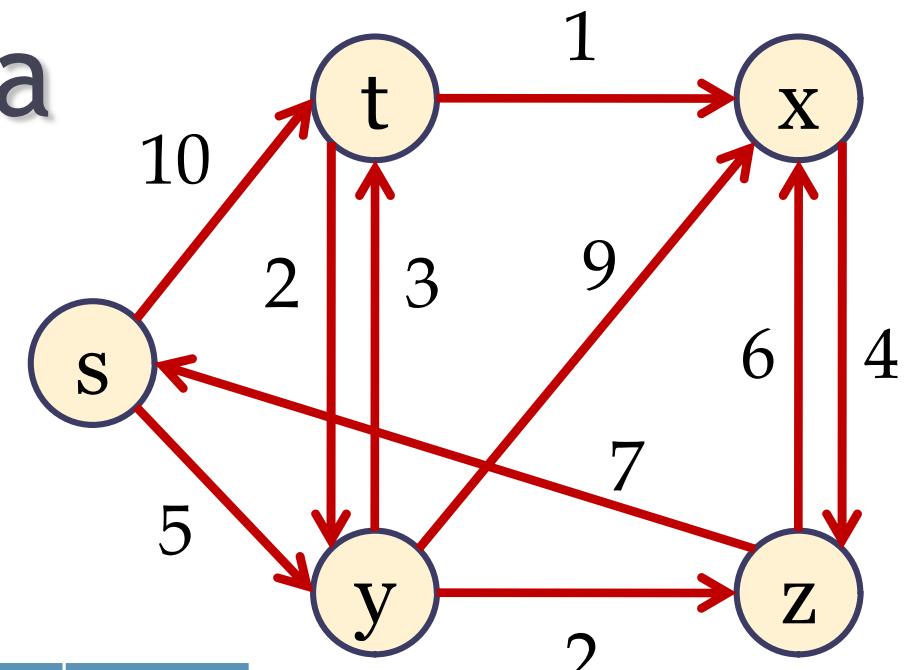
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

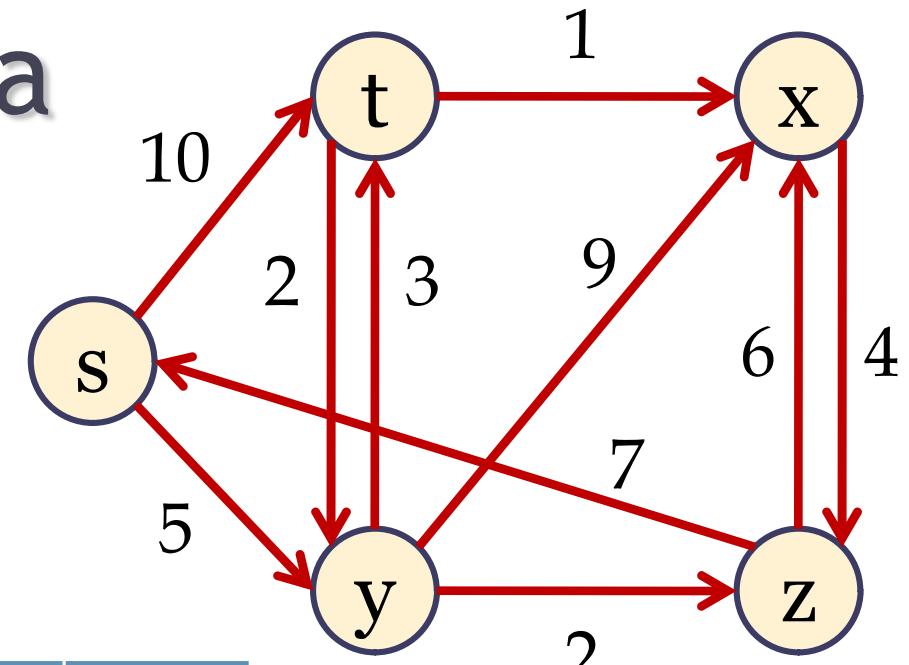
→ para cada $v \in Adj[u]$

→ relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

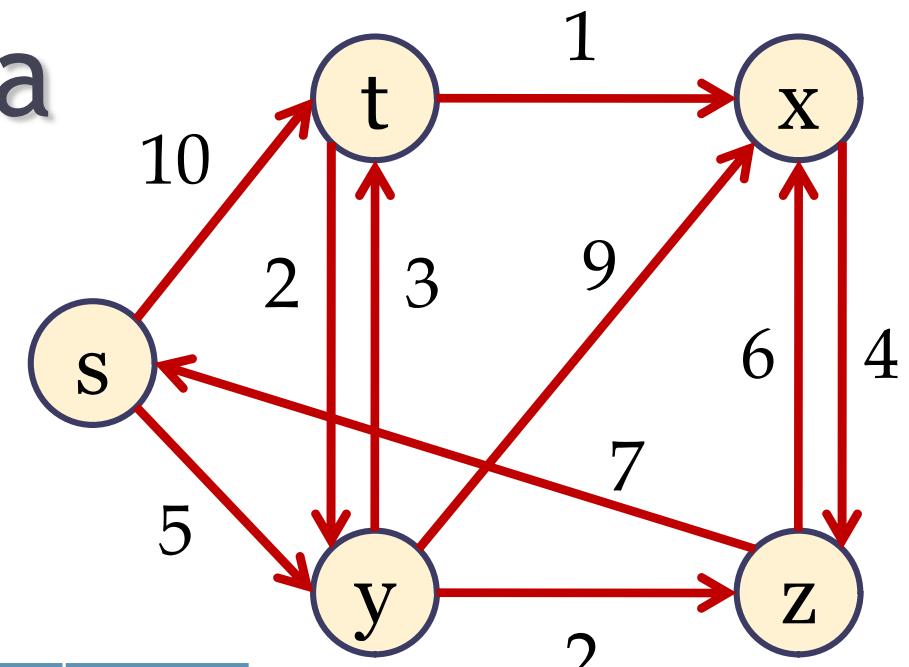
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	14	5	7
π	NULL	y	y	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

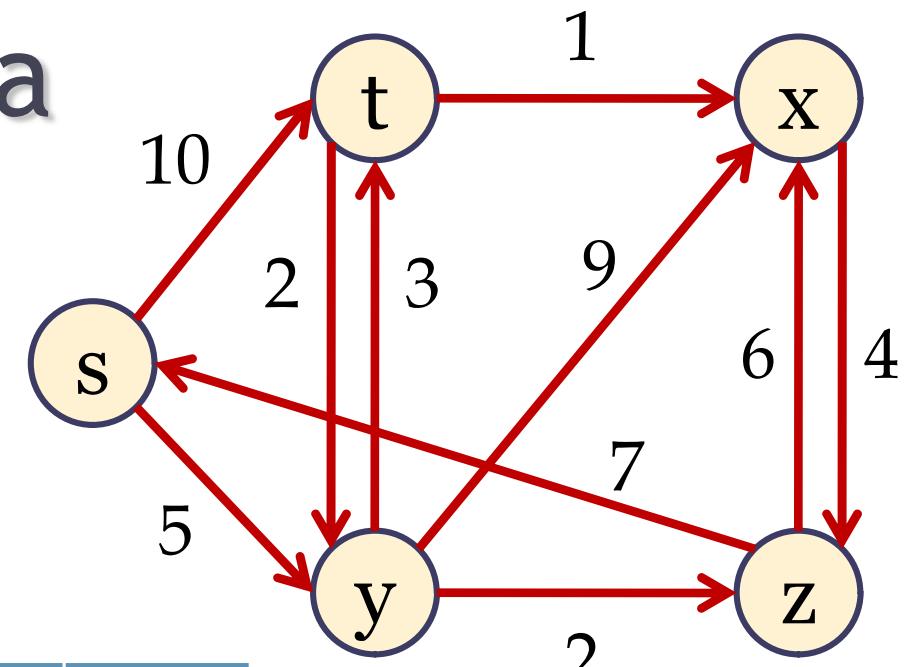
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

→ Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

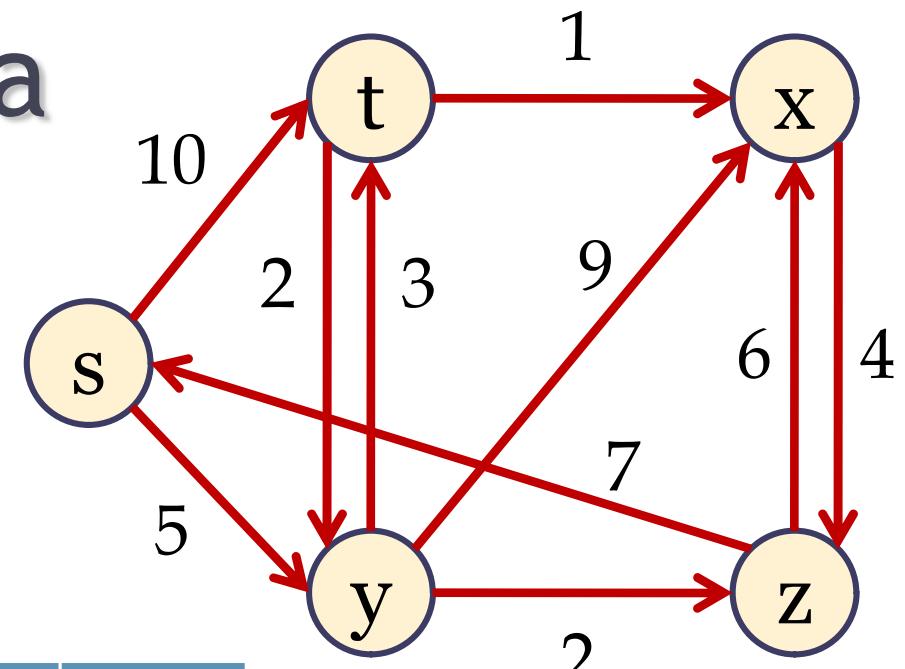
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

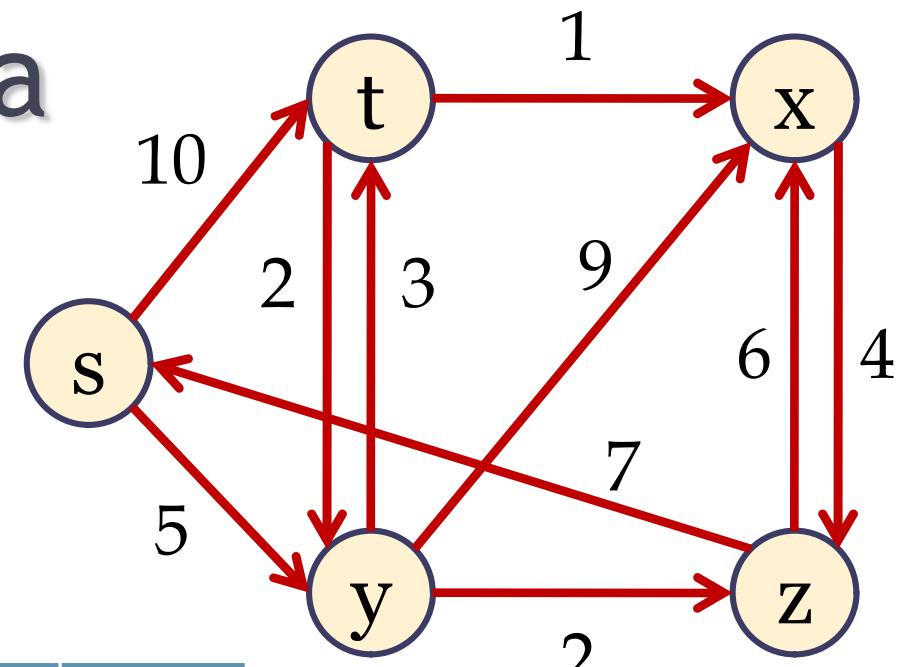
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

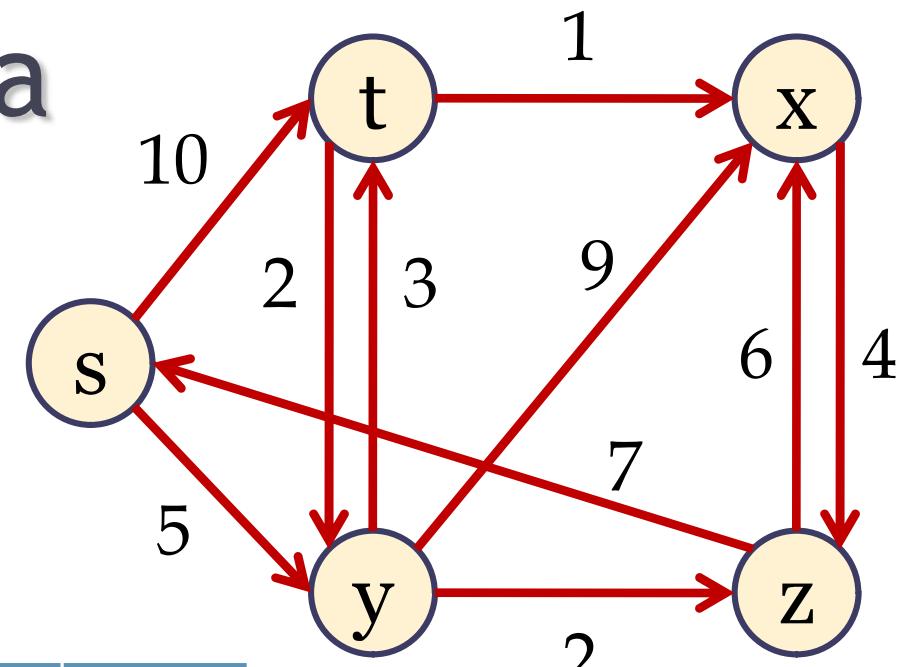
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	z

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	X	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow \text{extrair Mínino}(Q)$

$S \leftarrow S \cup \{u\}$

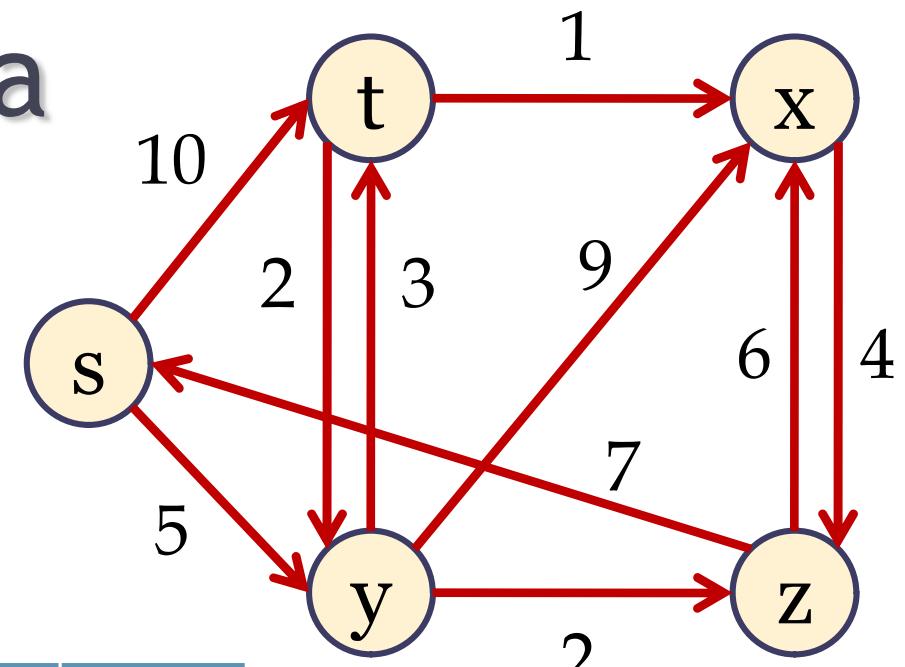
 para cada $v \in \text{Adj}[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	---	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

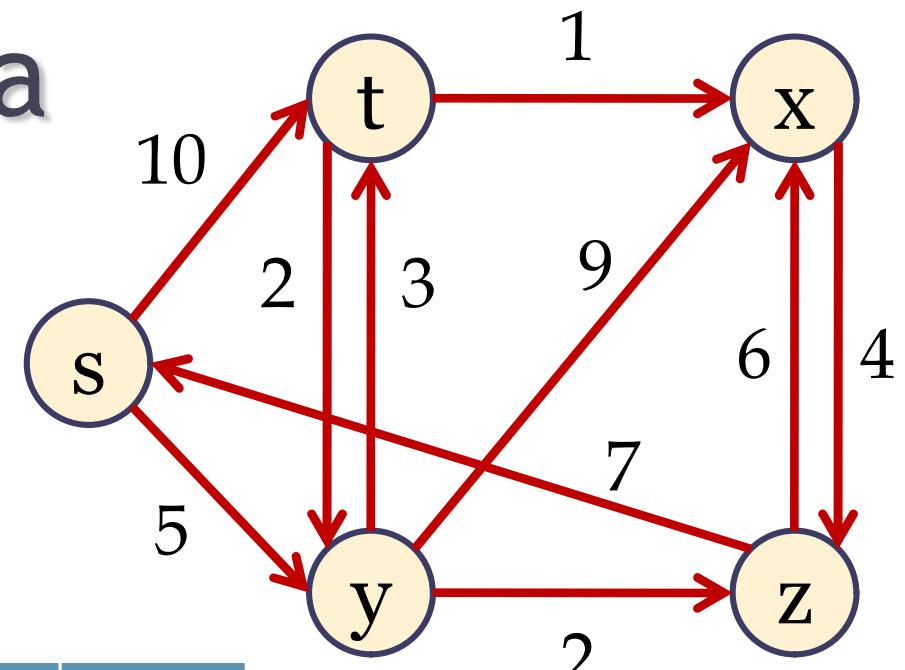
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	---	X	---	---
S	X	---	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

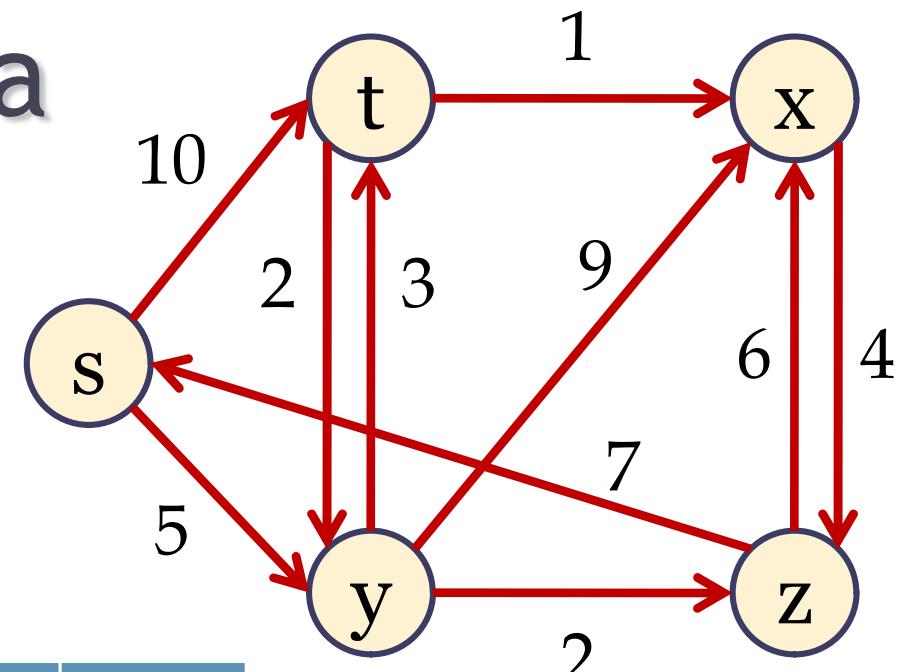
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

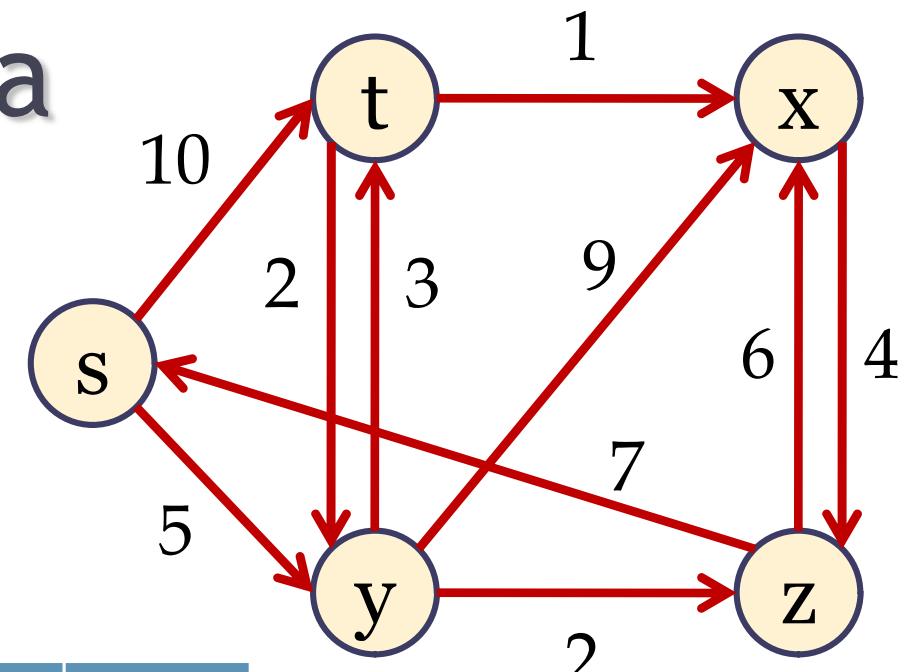
→ para cada $v \in Adj[u]$

→ relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

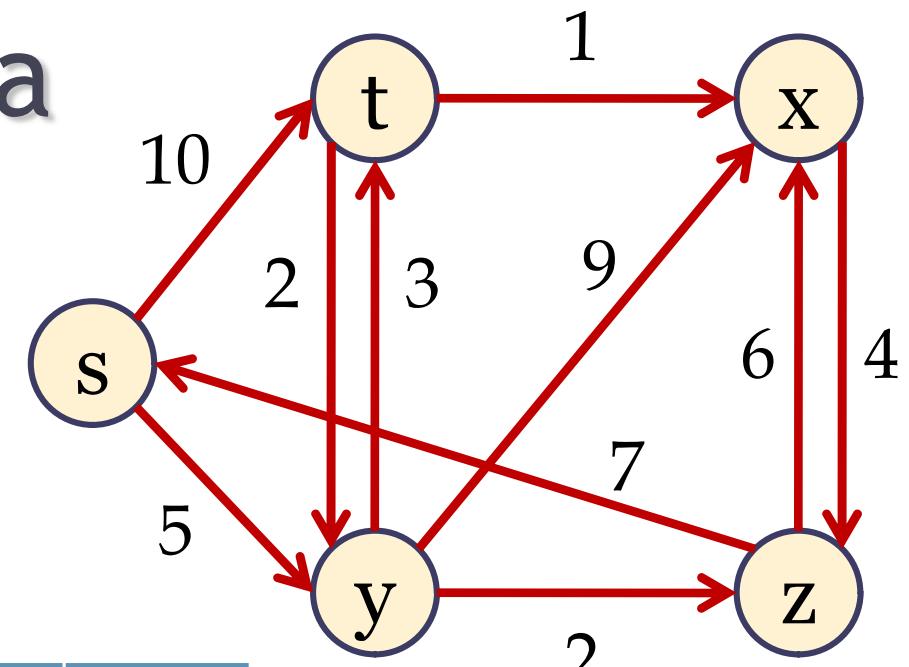
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	13	5	7
π	NULL	y	z	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

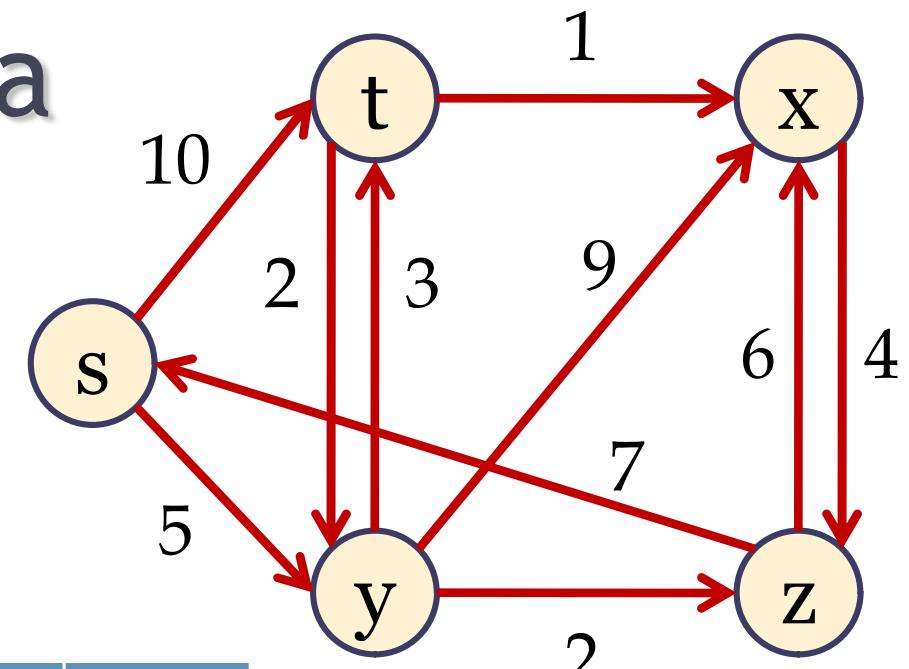
 → para cada $v \in Adj[u]$

 → relaxa(u, v, w)

 fim para

 fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

→ Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

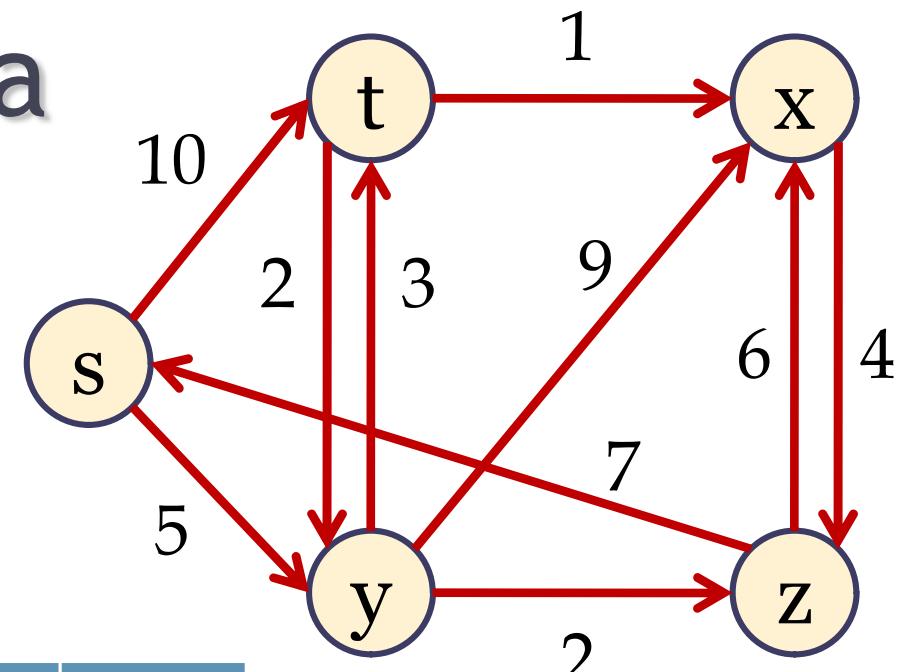
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

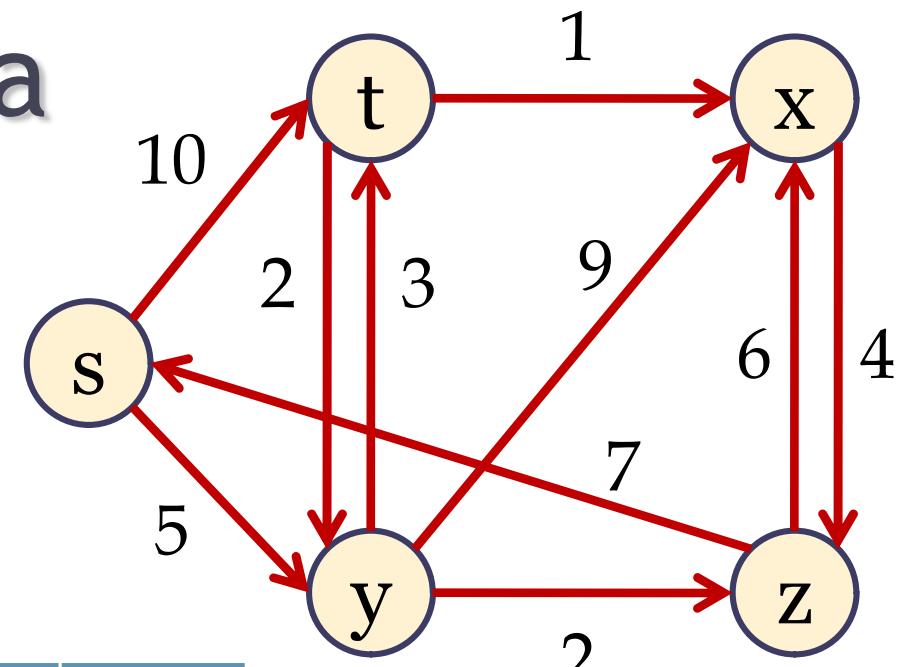
 para cada $v \in Adj[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$\rightarrow u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

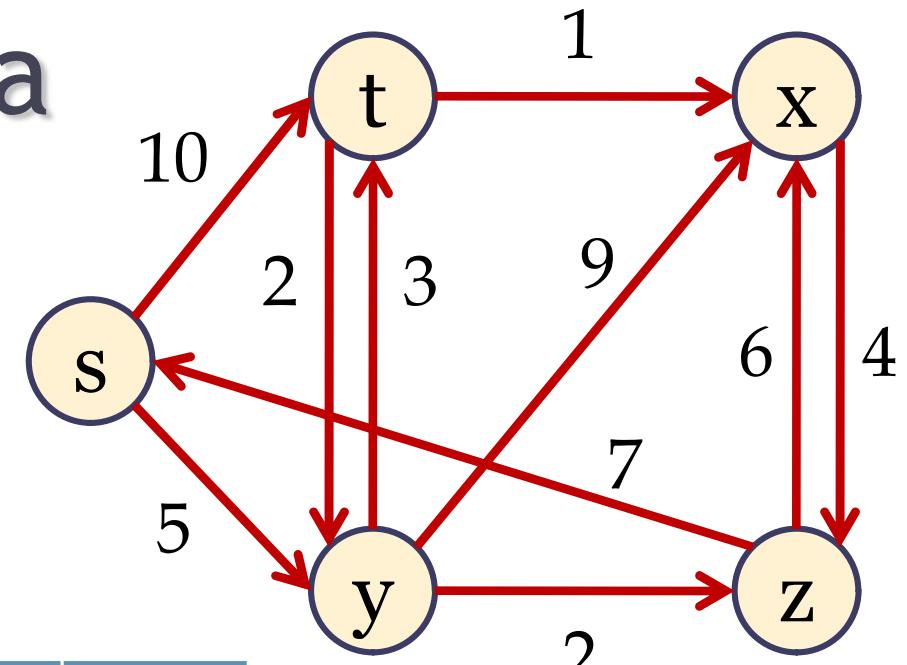
 para cada $v \in Adj[u]$

 relaxa(u, v, w)

 fim para

fim enquanto

fim



variável	valor
u	t

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	X	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

 → $u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

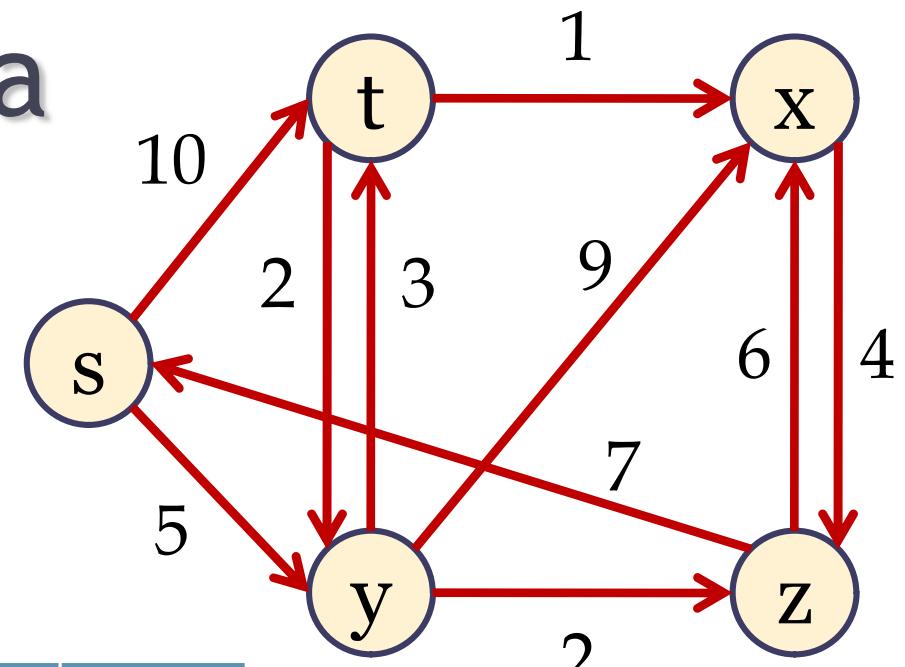
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

 $S \leftarrow S \cup \{u\}$

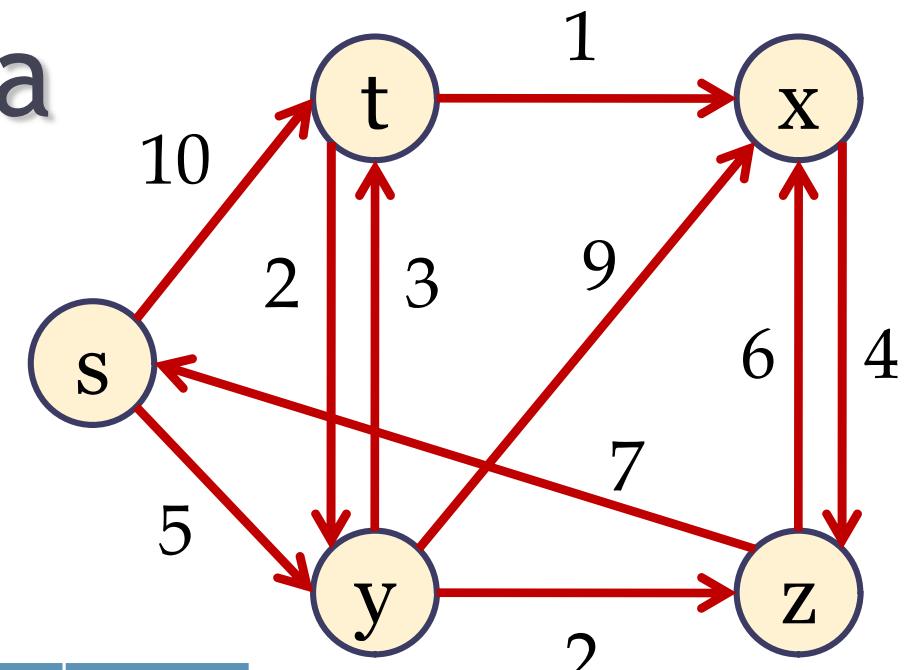
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	---	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

→ $S \leftarrow S \cup \{u\}$

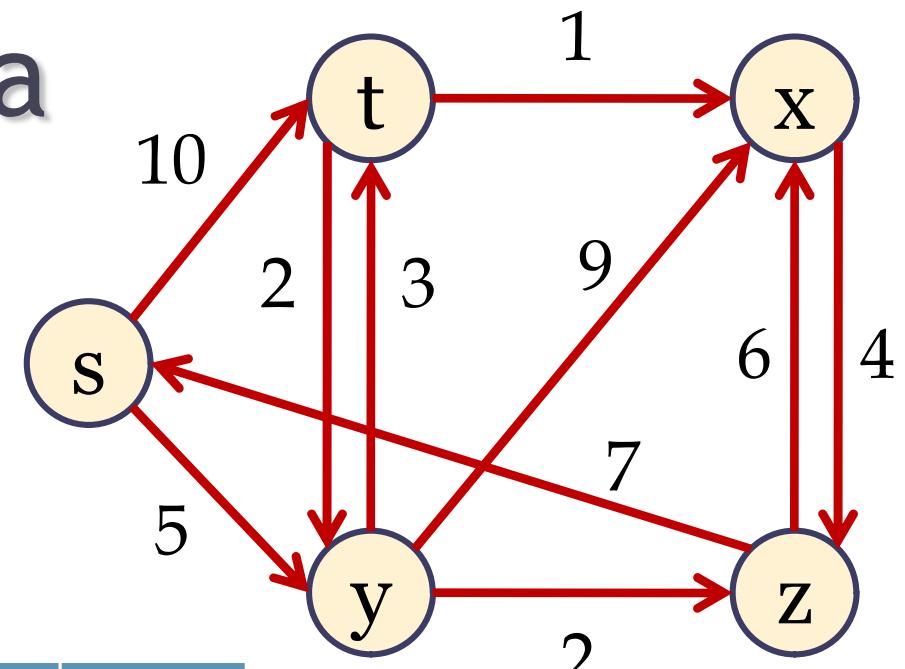
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	X	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

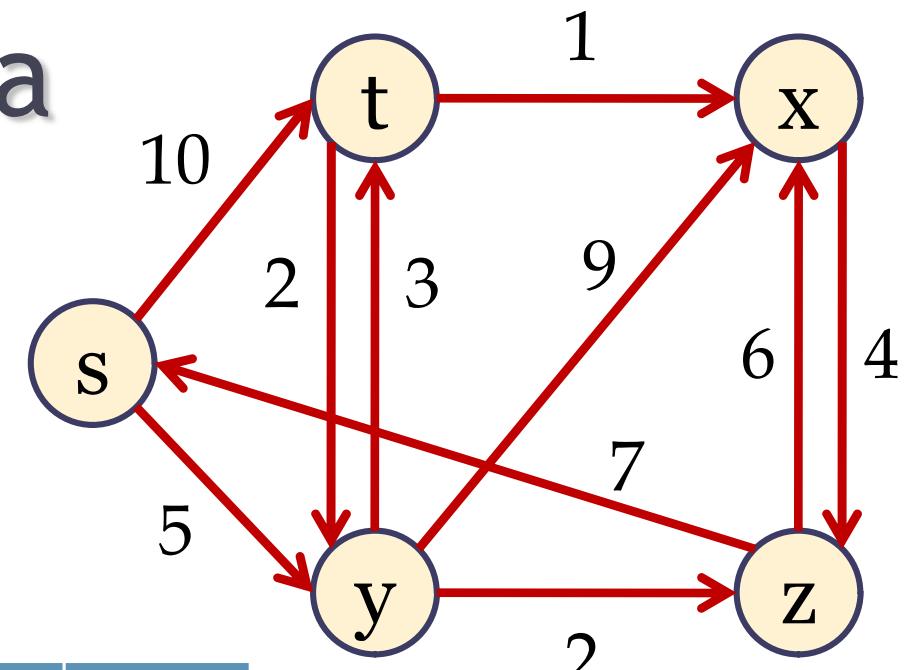
→ para cada $v \in Adj[u]$

→ relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	X	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

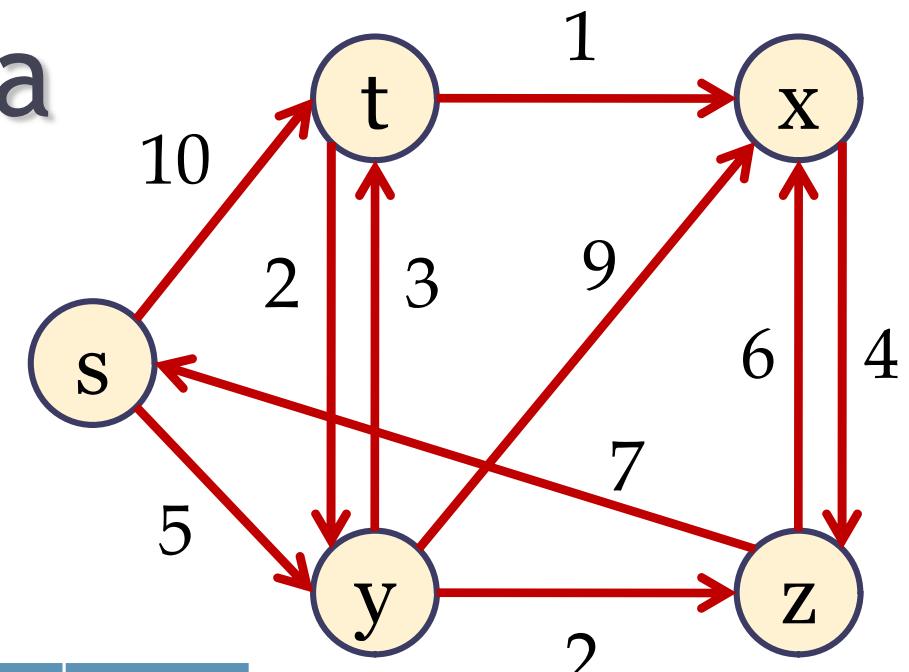
→ para cada $v \in Adj[u]$

→ relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	X	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

→ Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

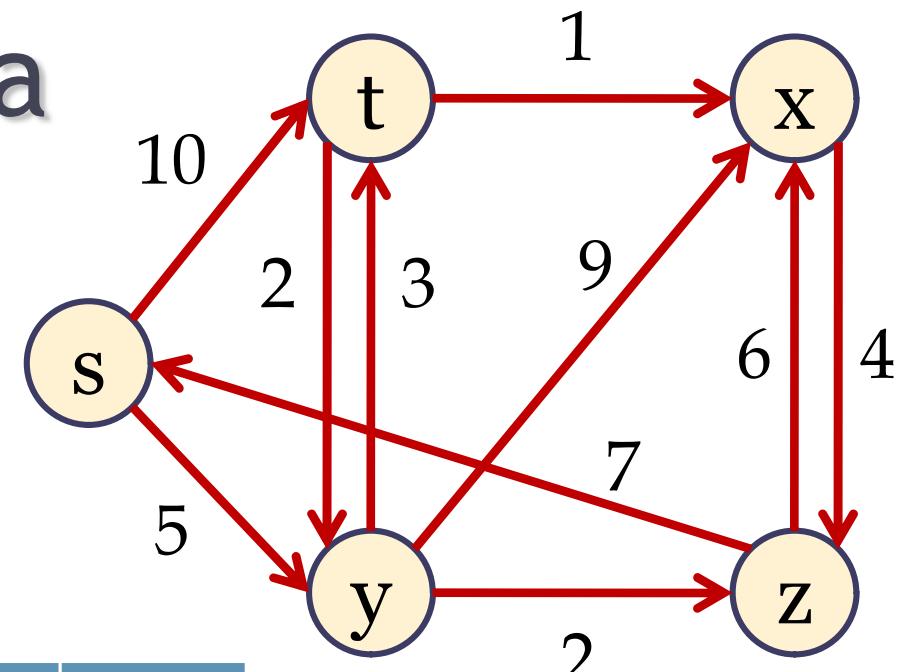
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

fim



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	X	X	X

Algoritmo de Dijkstra

DIJSKSTRA($G = (V, A), w, s$)

INICIALIZA(G, s)

$S \leftarrow \{ \}$

$Q \leftarrow V$

Enquanto $|Q| \neq 0$

$u \leftarrow$ extrair M inino(Q)

$S \leftarrow S \cup \{u\}$

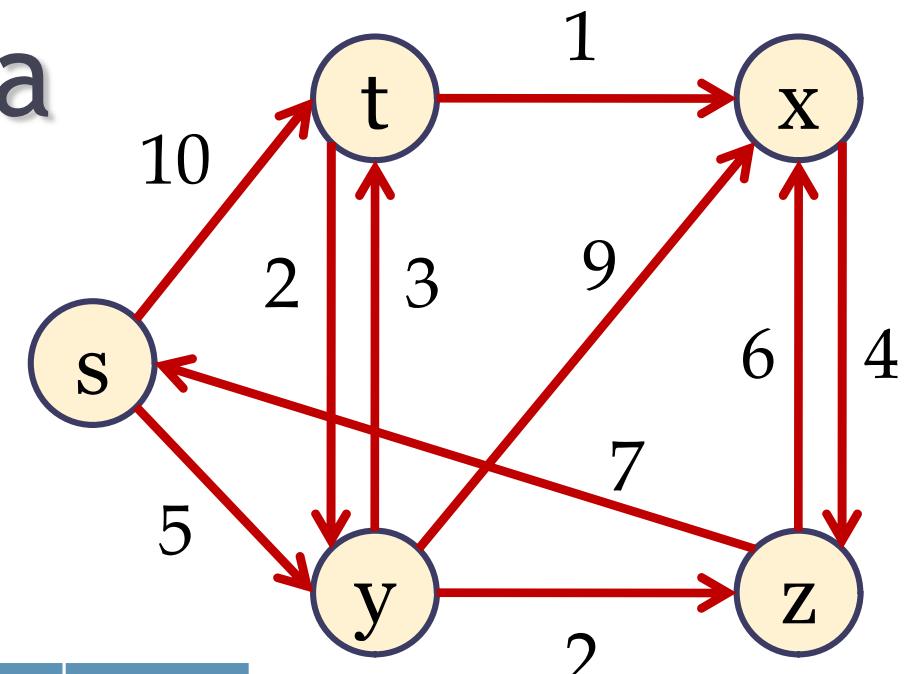
para cada $v \in Adj[u]$

relaxa(u, v, w)

fim para

fim enquanto

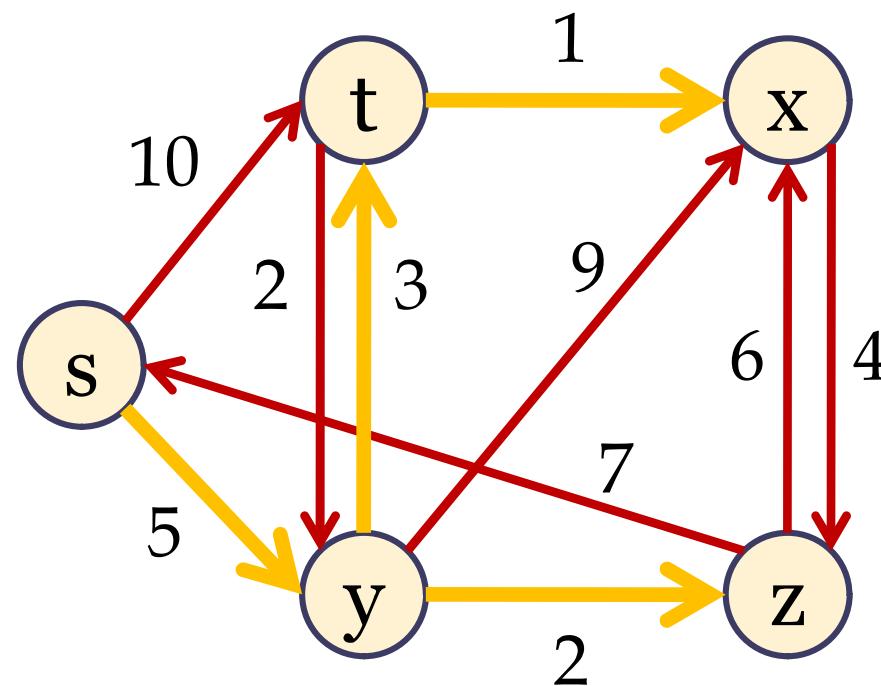
fim 



variável	valor
u	x

vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y
Q	---	---	---	---	---
S	X	X	X	X	X

Algoritmo de Dijkstra



vértice	s	t	x	y	z
d	0	8	9	5	7
π	NULL	y	t	s	y

Algoritmo de Dijkstra

- Reflexão dos alunos...
 - E se o grafo possuir arestas de peso negativo?
 - A resposta do algoritmo de Dijkstra é correta?
 - Como otimizar o algoritmo de Dijkstra para localizar o menor caminho entre um par de vértices? Ele pode ser otimizado?

Dijkstra

- (1930-2002)
- Site: <http://www.cs.utexas.edu/users/EWD/>
- No site você encontra toda a produção de publicada oficialmente de Dijkstra, incluindo também inúmeros manuscritos com provas e teorias nas mais diversas áreas da ciência da computação...

