Grafos: caminhos mínimos Parte 2

SCC0216 Modelagem Computacional em Grafos

Thiago A. S. Pardo
Maria Cristina F. Oliveira

Relembrando...

- Caminho mínimo
  - Variações de origem e destino
  - Pesos/custos
  - Ciclos
- Algoritmo de Dijkstra
  - Um dos mais conhecidos e utilizados
  - O(|A| log|V|), se bem implementado

Algoritmo de **Bellman-Ford**

- Características
  - Caminhos mais curtos de origem única
  - Arestas podem ter peso negativo
  - Detecta ciclos negativos

- Método
  1. Parte de estimativas pessimistas para cada vértice
  2. Faz o relaxamento de **todas** as arestas |V| - 1 vezes, garantindo que os caminhos mínimos prevaleçam

Caminhos mínimos

- Sub-rotina para relaxamento de arestas

relax(u, v, w) // (u,v) é a aresta, w é o seu peso

início

se d[v] > d[u] + w(u,v) **então**

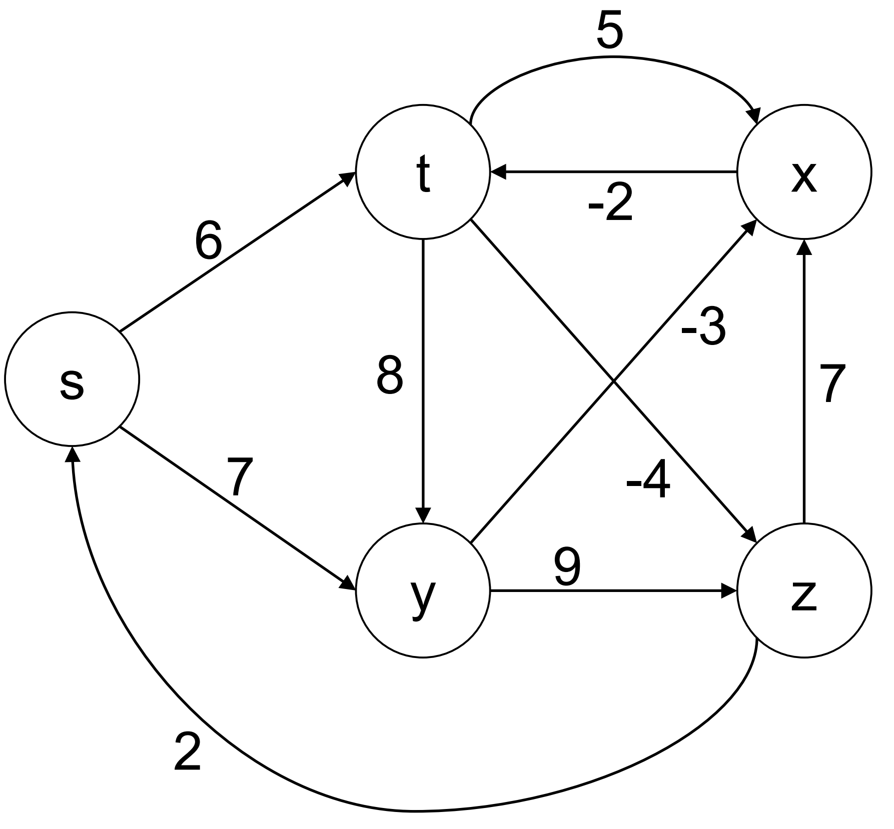
d[v] = d[u] + w(u,v)

antecessor[v]=u

fim

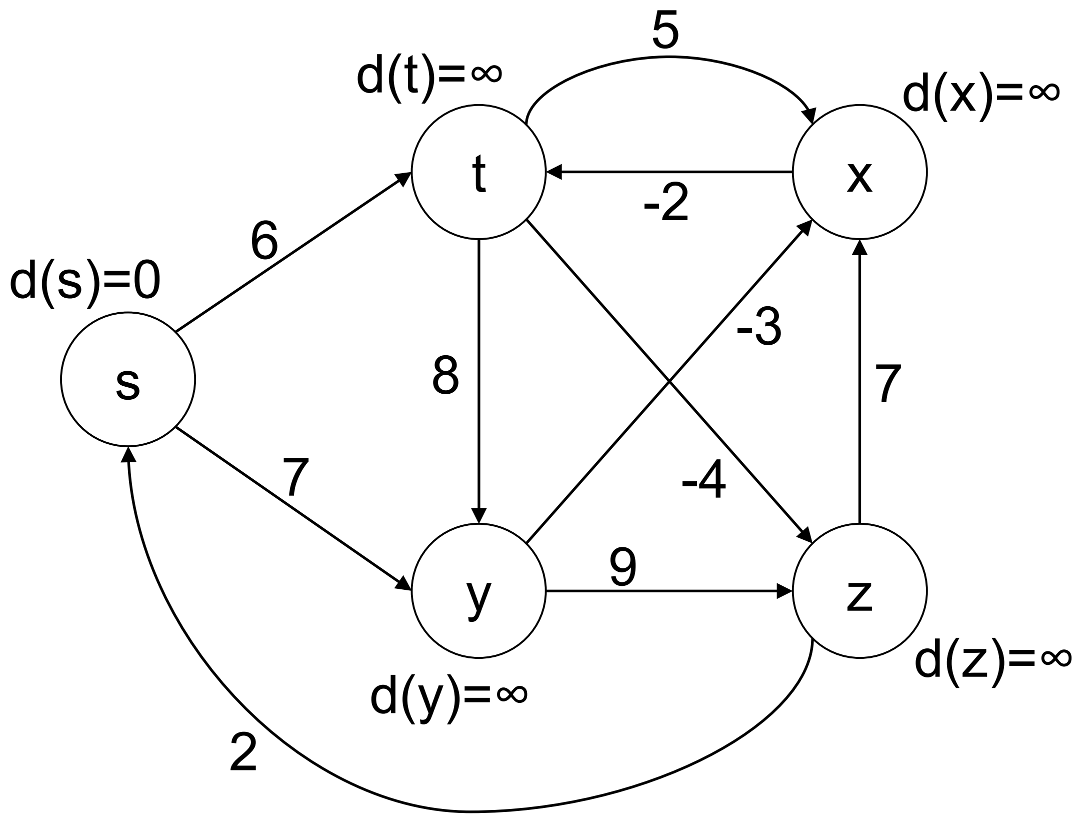
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)



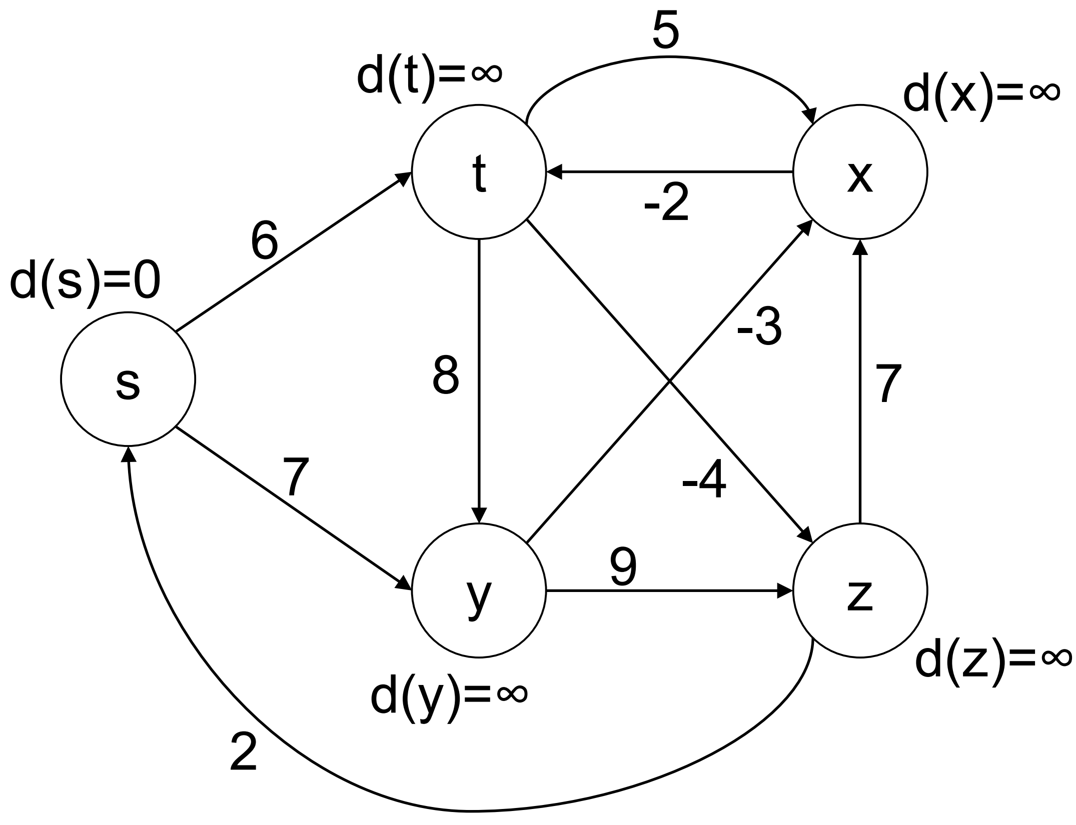
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Estimativas pessimistas



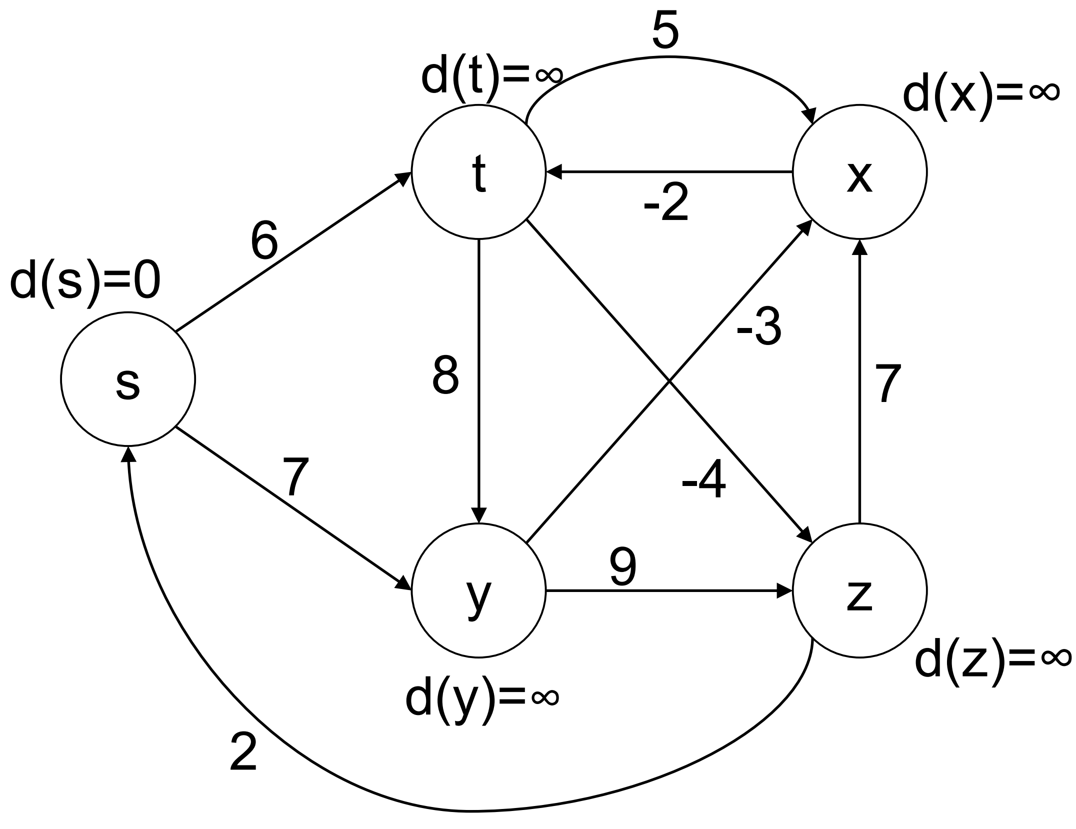
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Ordem (aleatória) das arestas
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



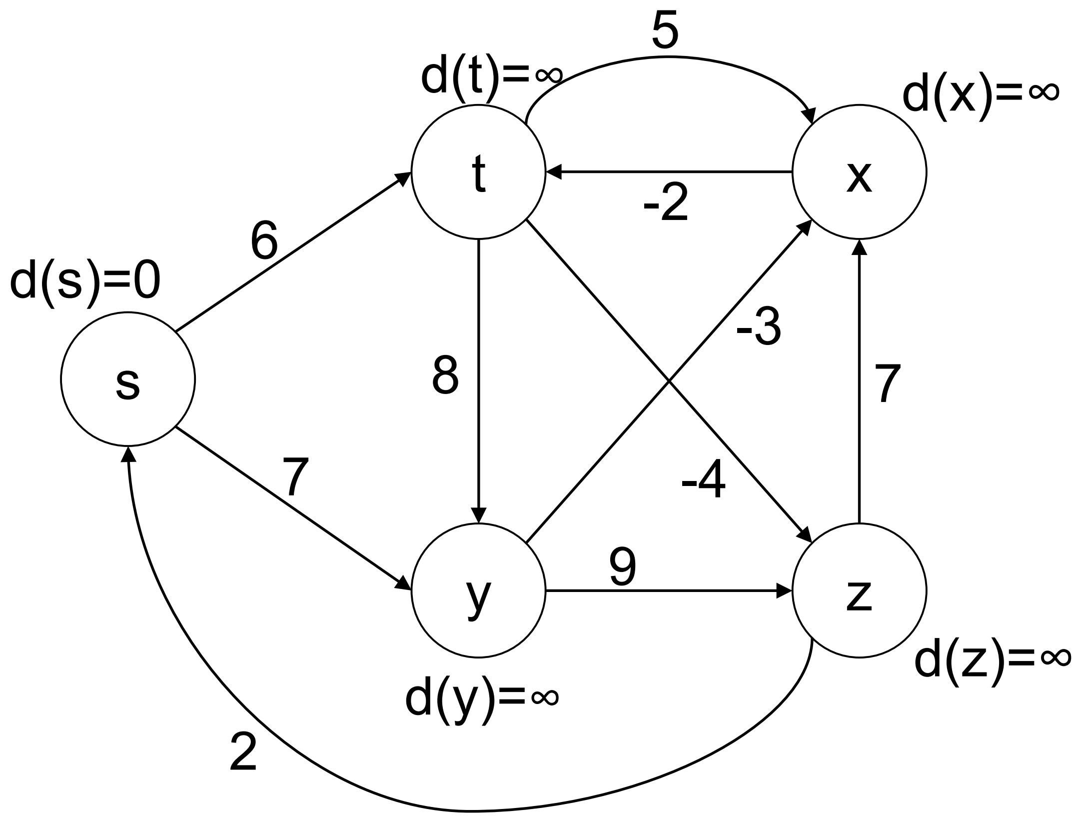
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 1 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



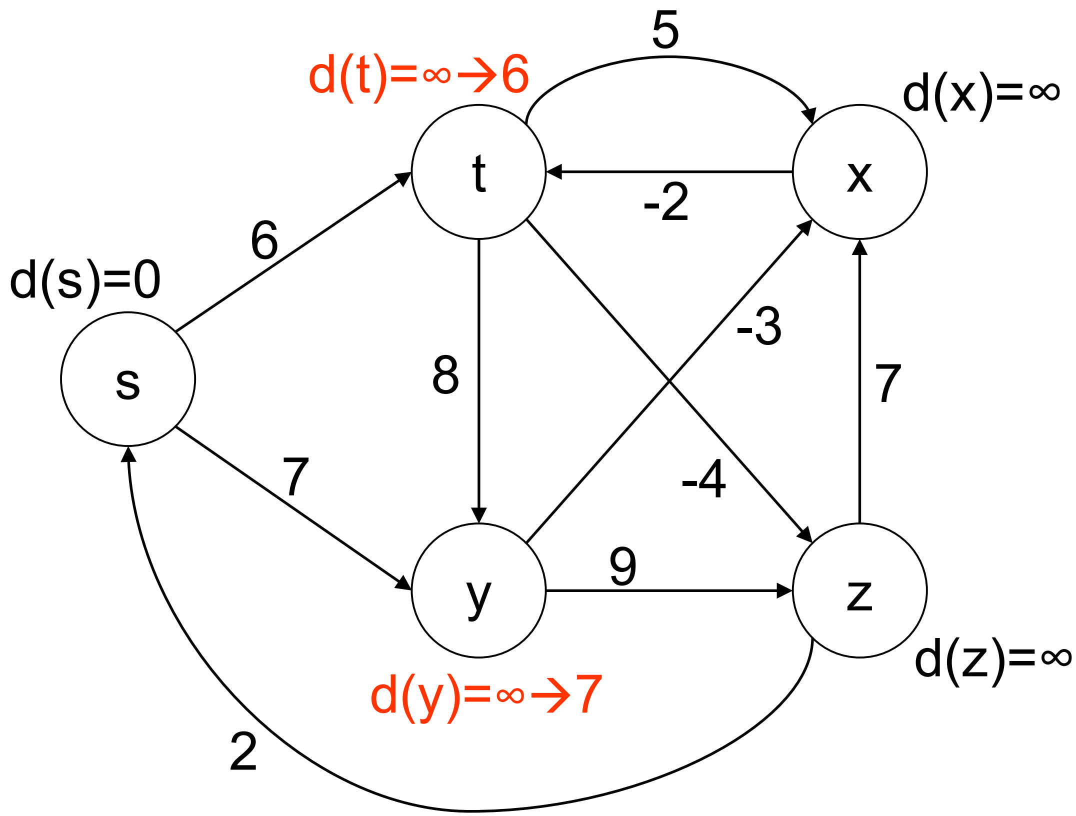
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 1 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



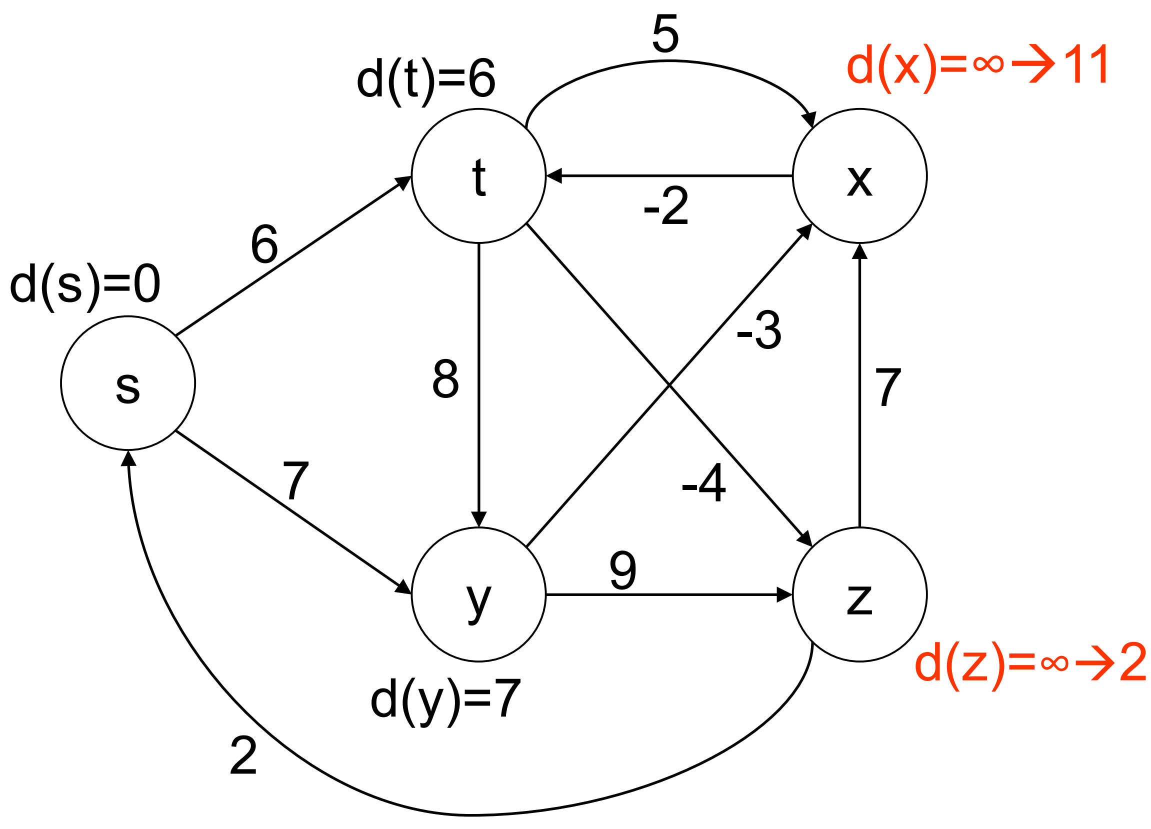
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 1 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



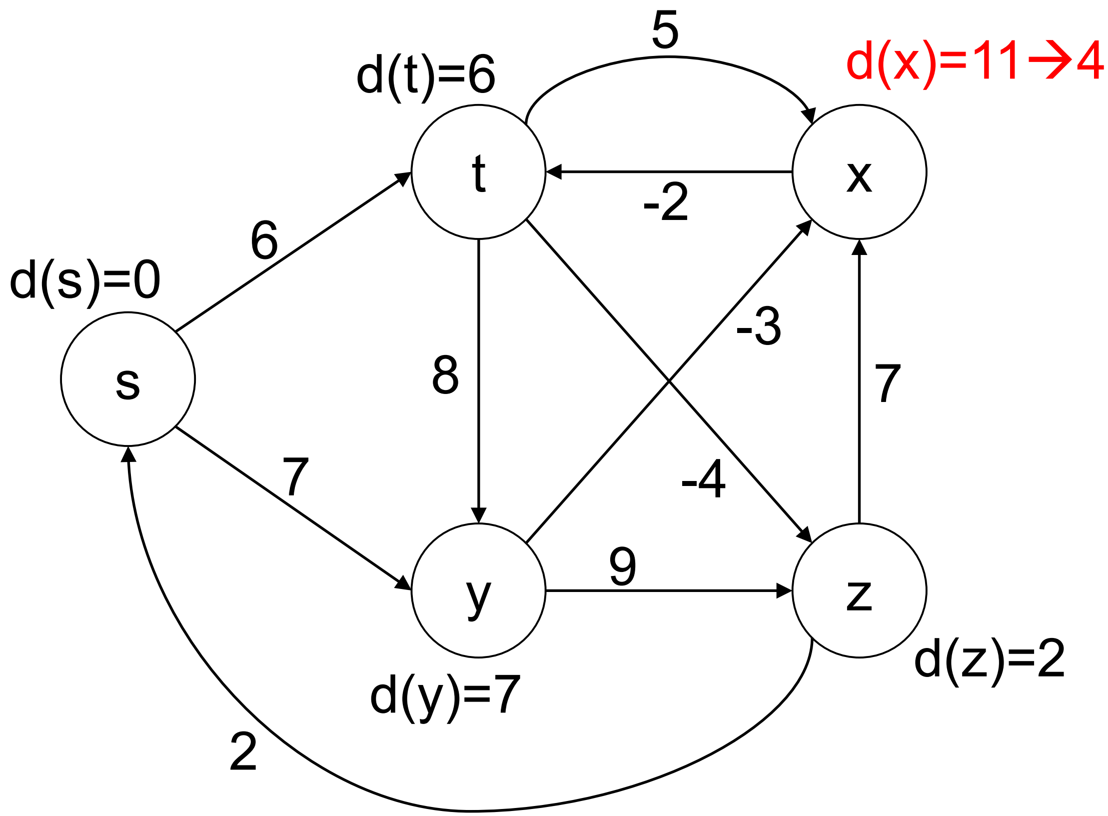
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 2 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



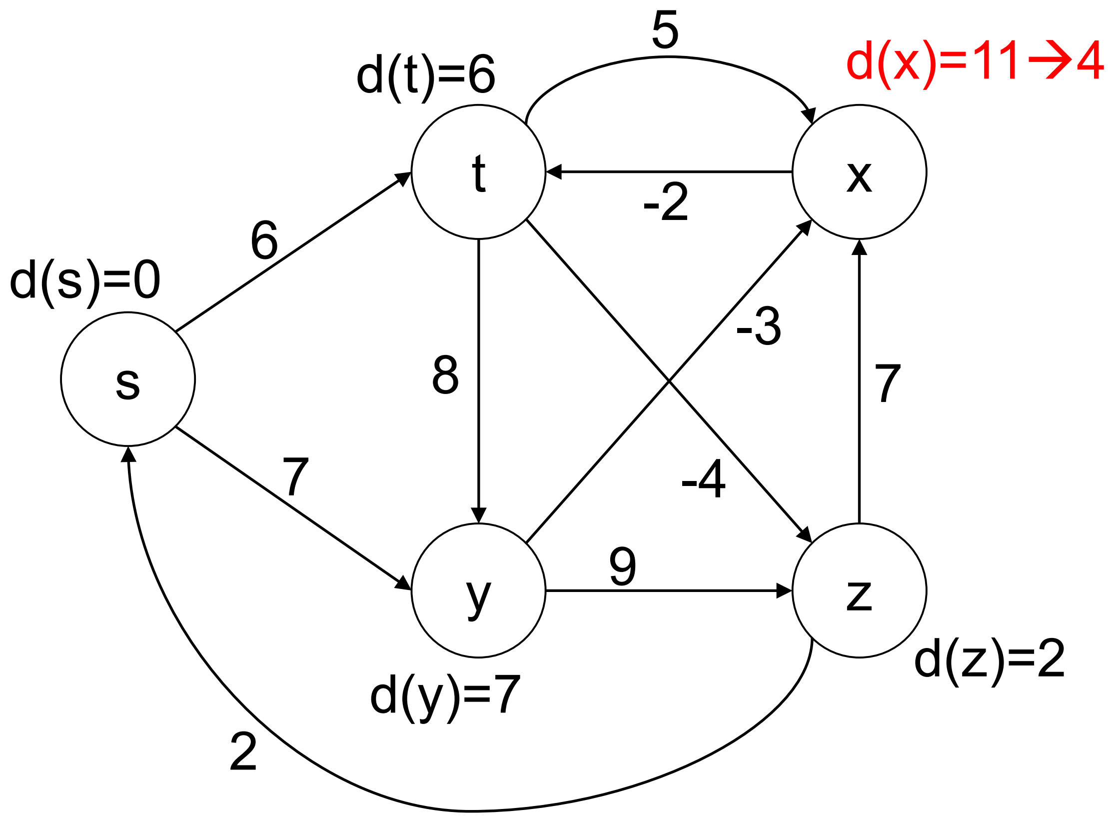
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 2 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



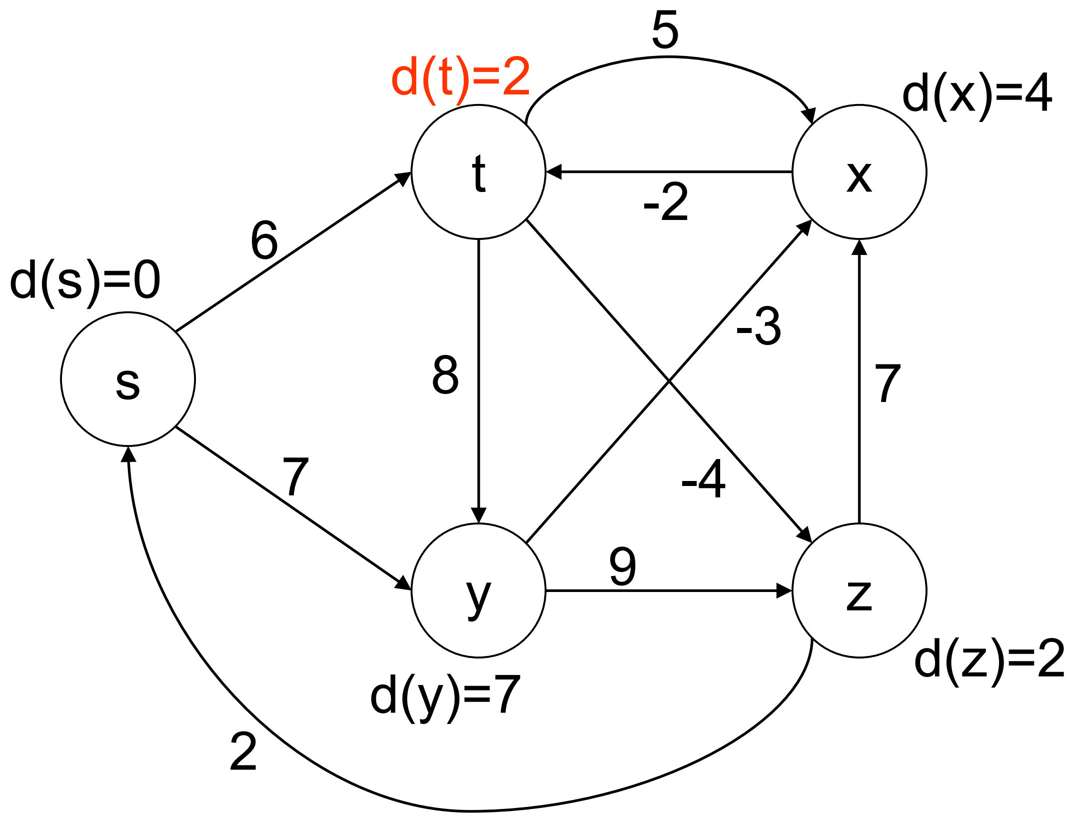
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 2 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



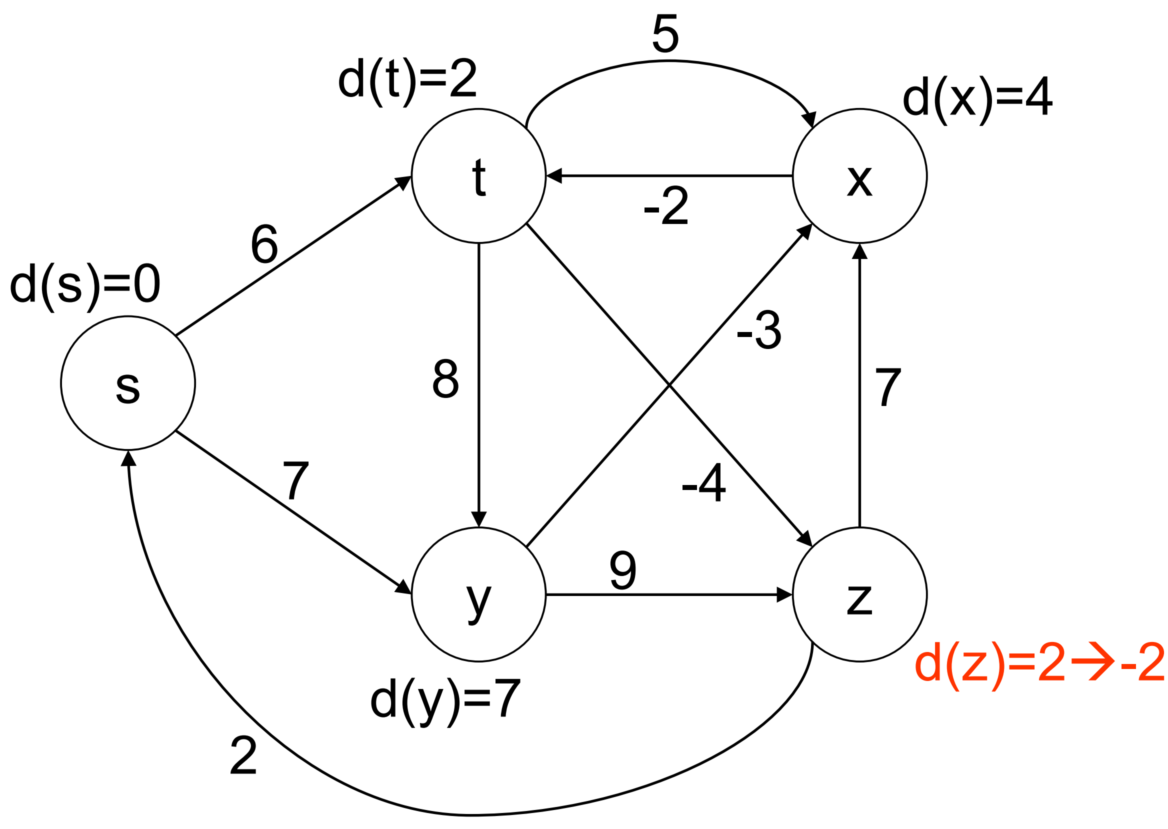
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 3 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



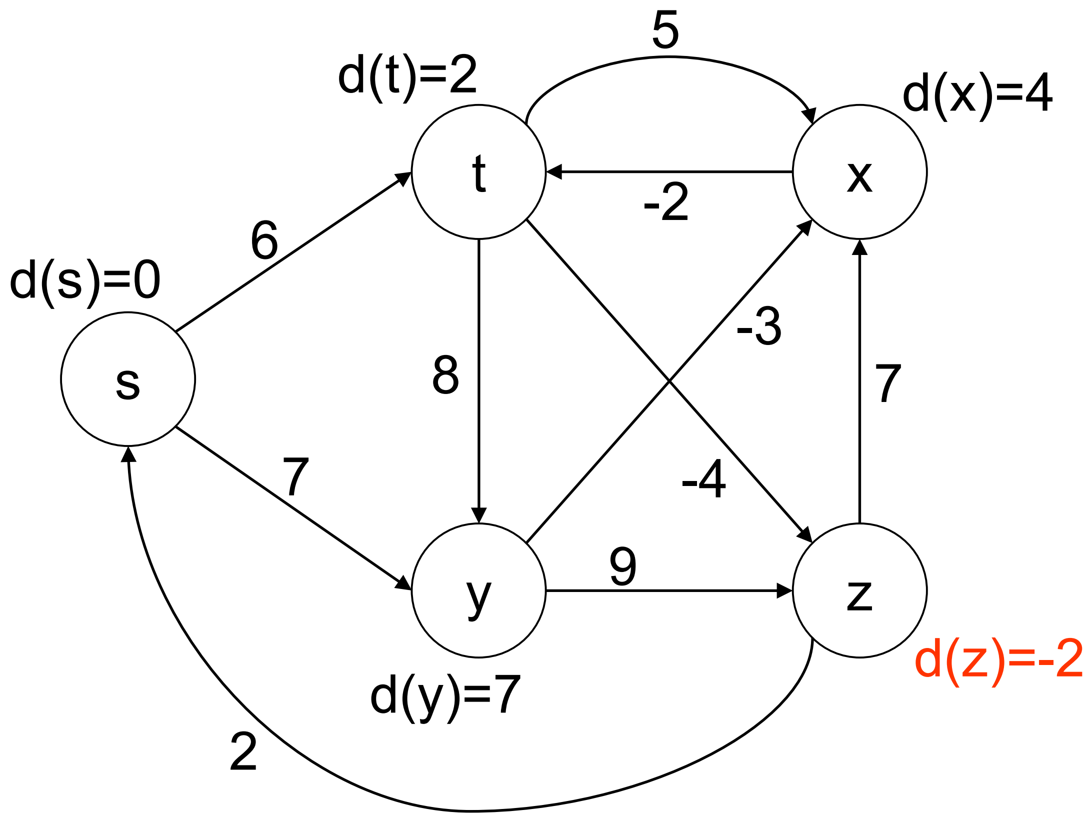
Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 4 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



Algoritmo de Bellman-Ford

- Exemplo (a partir de s)
  - Rodada 4 (de |V|-1=4) de relaxamento
    - (t,x), (t,y), (t,z)
    - (x,t)
    - (y,x), (y,z)
    - (z,x), (z,s)
    - (s,t), (s,y)



Algoritmo de Bellman-Ford

- Atenção
  - Se aresta continuamente diminuir, há ciclo cuja soma de pesos é negativa
    - Não há caminho “mais curto”, pois sempre é possível “baratear” mais o caminho
  - Se em uma iteração não houver alteração de custos, não é preciso continuar
    - Algoritmo já convergiu!
  - Método mais versátil do que Dijkstra
    - Porém, também mais caro

Algoritmo de Bellman-Ford

- Na implementação, uso do <u>antecessor</u>
  - Sempre que ocorre um relaxamento de aresta (*u*,*v*), o antecessor é armazenado
    - antecessor[v]=u
  - Para saber o caminho mínimo da origem *s* até qualquer vértice *v*, basta rastrear o antecessor de *v* até chegar em *s*

Algoritmo de Bellman-Ford

```
BELLMAN-FORD(G, w, s)
início
    //inicializa variáveis
    para cada vértice v faça
            d[v]=∞
            antecessor[v]=-1
    d[s]=0
    //faz relaxamento de arestas e determina caminhos mais curtos
    para i=1 até |V|-1 faça
            para cada aresta (u,v) faça     // w é o peso da aresta (u,v)
                    relax(u,v,w)
    //verifica se há ciclos negativos
    para cada aresta (u,v) faça
            se d[v] > d[u] + w(u,v) então
                    retorna "HÁ CICLO DE PESO NEGATIVO"
    retorna "NÃO HÁ CICLO DE PESO NEGATIVO"
fim
```

Algoritmo de Bellman-Ford

- Complexidade de tempo: ?

Algoritmo de Bellman-Ford

- Complexidade de tempo: O(|V| x |A|)
  - Por que?

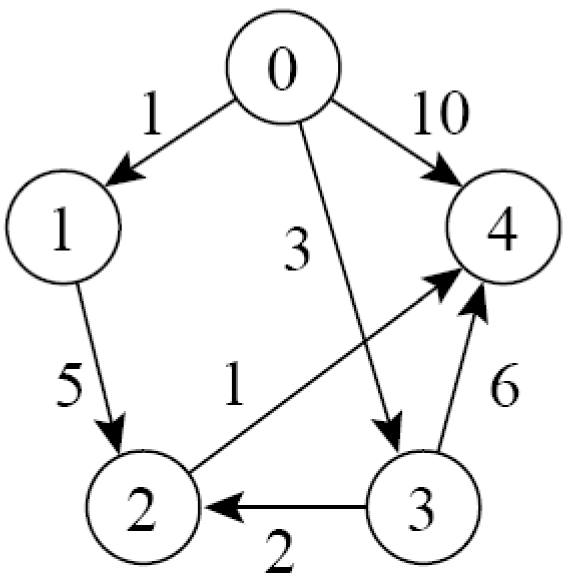
Algoritmo de Bellman-Ford

```
BELLMAN-FORD(G, w, s)
    início
    //inicializa variáveis
    para cada vértice v faça                    // O(|V|)
        d[v]=∞
        antecessor[v]=-1
    d[s]=0
    //faz relaxamento de arestas e determina caminhos mais curtos
    para i=1 até |V|-1 faça                     // O(|V|)
        para cada aresta (u,v) faça             // O(|A|)
            relax(u,v,w)
    //verifica se há ciclos negativos
    para cada aresta (u,v) faça                 // O(|A|) no total
        se d[v] > d[u] + w(u,v) então
            retorna "HÁ CICLO DE PESO NEGATIVO"
    retorna "NÃO HÁ CICLO DE PESO NEGATIVO"
    fim
```

Exercícios

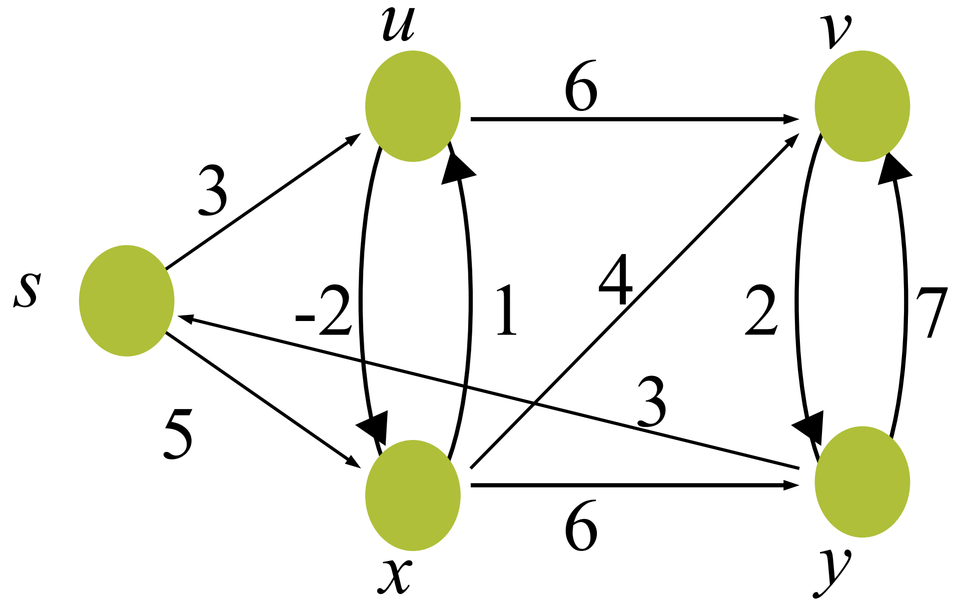
Exercício

- Calcule os caminhos mínimos para o grafo abaixo a partir do vértice 0 aplicando o algoritmo de Bellman-Ford



Exercício

- Calcule os caminhos mínimos para o grafo abaixo a partir do vértice s aplicando o algoritmo de Bellman-Ford



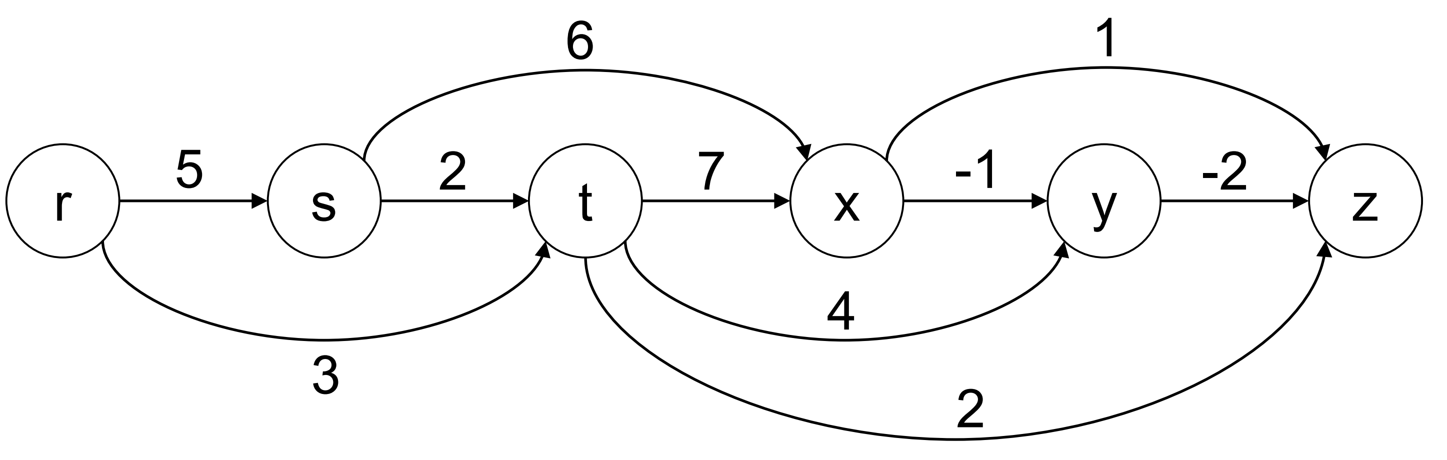
Algoritmo baseado na **ordenação topológica**

- Características
  - Caminho mais curto de origem única
  - Grafos sem ciclos
  - Admite pesos negativos

- Método
  - Faz-se a ordenação topológica do grafo
  - Percorre-se a lista de vértices na sequência topológica, relaxando-se todas as arestas que partem de cada vértice

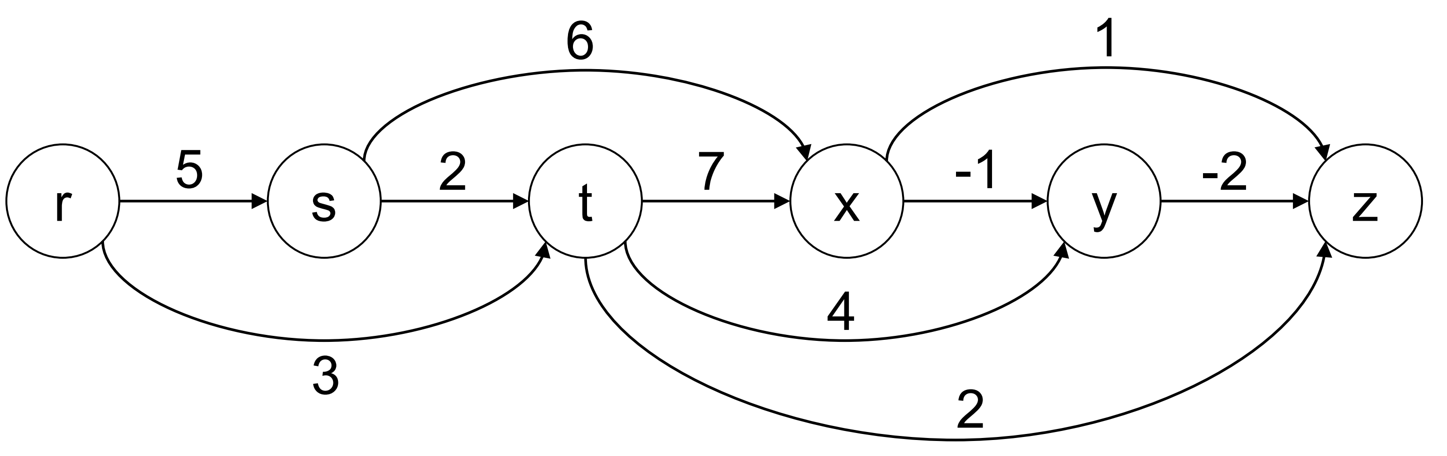
Algoritmo baseado na ordenação topológica

- Exemplo (a partir de s)



Algoritmo baseado na ordenação topológica

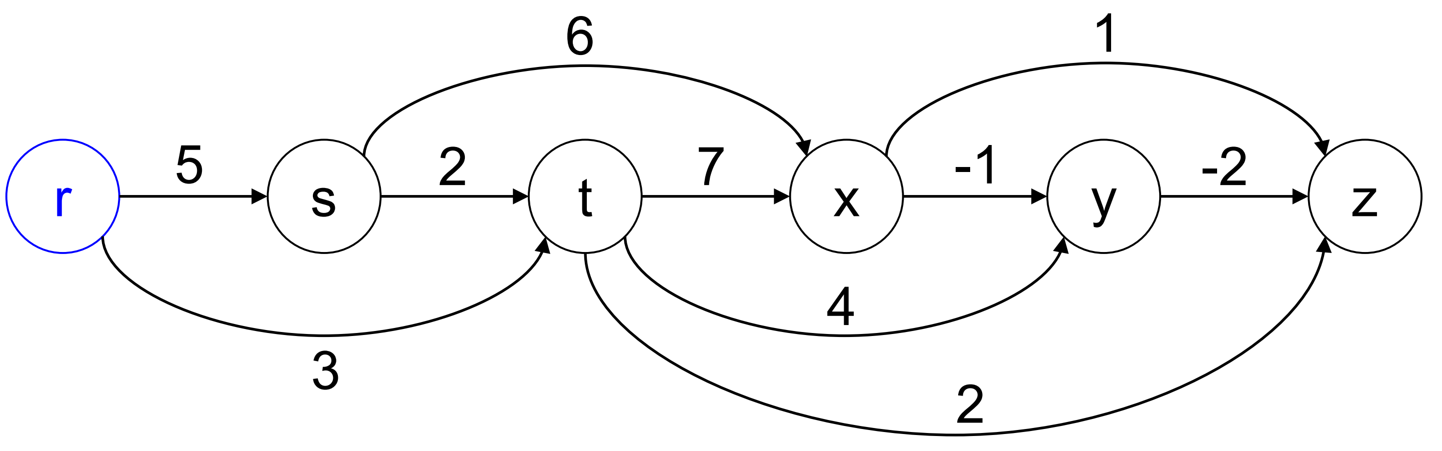
- Exemplo (a partir de s)
  - Estimativas pessimistas



d(r)=∞ d(s)=0 d(t)=∞ d(x)=∞ d(y)=∞ d(z)=∞

Algoritmo baseado na ordenação topológica

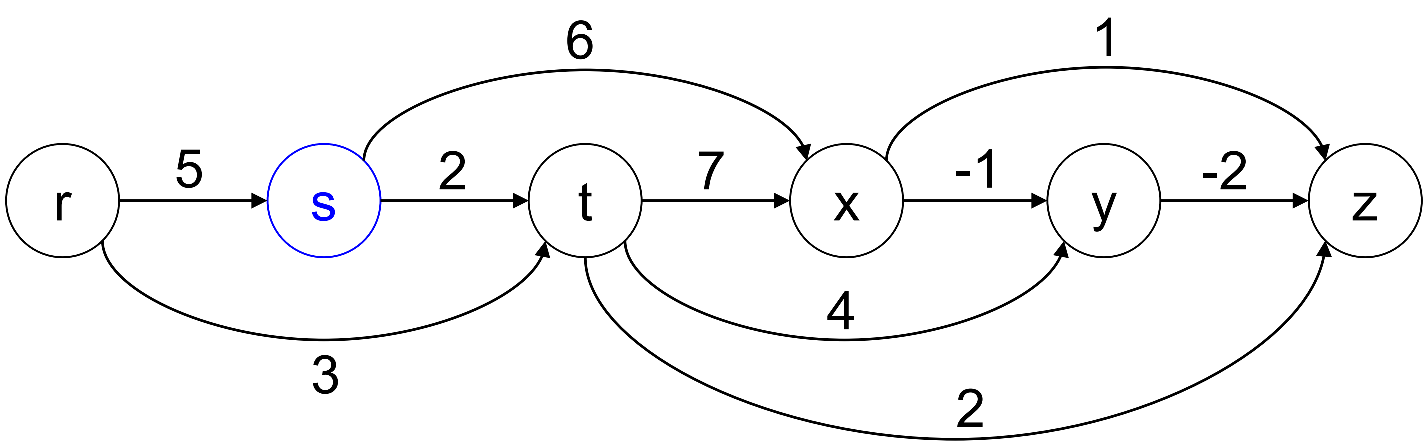
- Exemplo (a partir de s)
  - Arestas adjacentes a r



d(r)=∞ d(s)=0 d(t)=∞ d(x)=∞ d(y)=∞ d(z)=∞

Algoritmo baseado na ordenação topológica

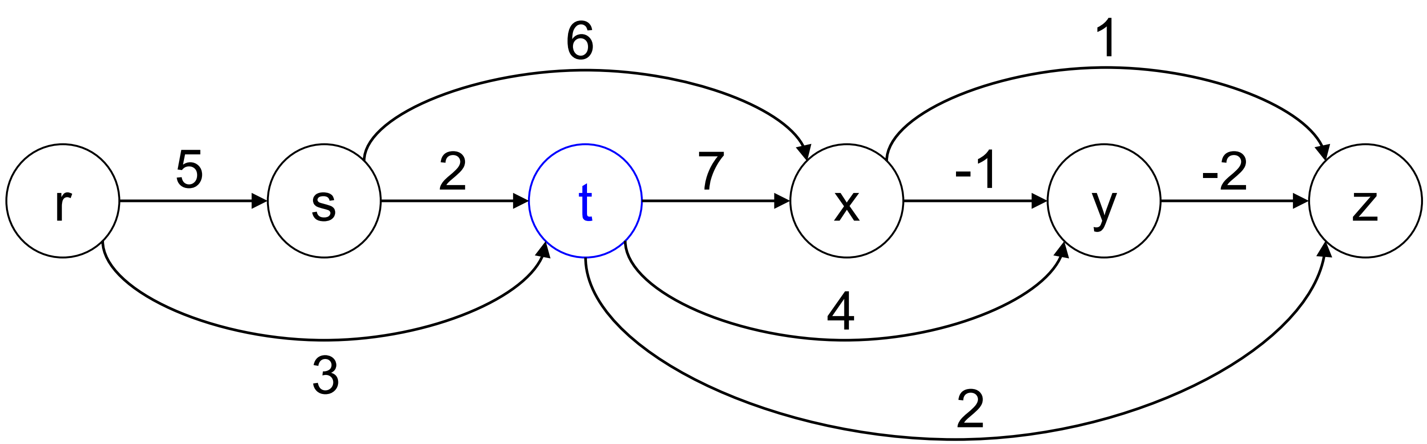
- Exemplo (a partir de s)
  - Arestas adjacentes a s



d(r)=∞ d(s)=0 d(t)=∞→2 d(x)=∞→6 d(y)=∞ d(z)=∞

Algoritmo baseado na ordenação topológica

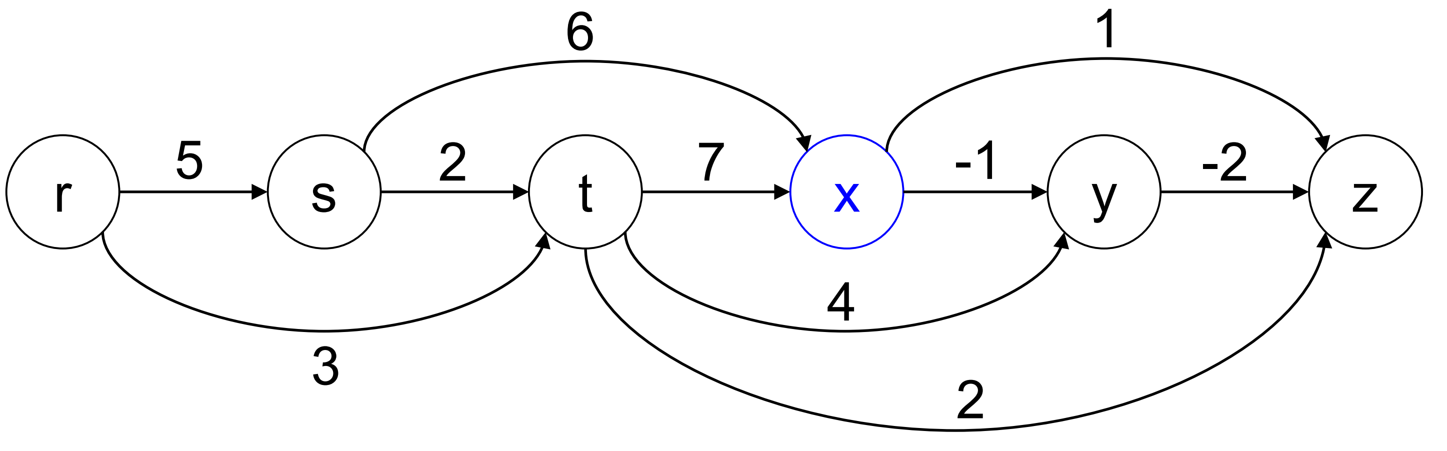
- Exemplo (a partir de s)
  - Arestas adjacentes a t



d(r)=∞ d(s)=0 d(t)=2 d(x)=6 d(y)=∞→6 d(z)=∞→4

Algoritmo baseado na ordenação topológica

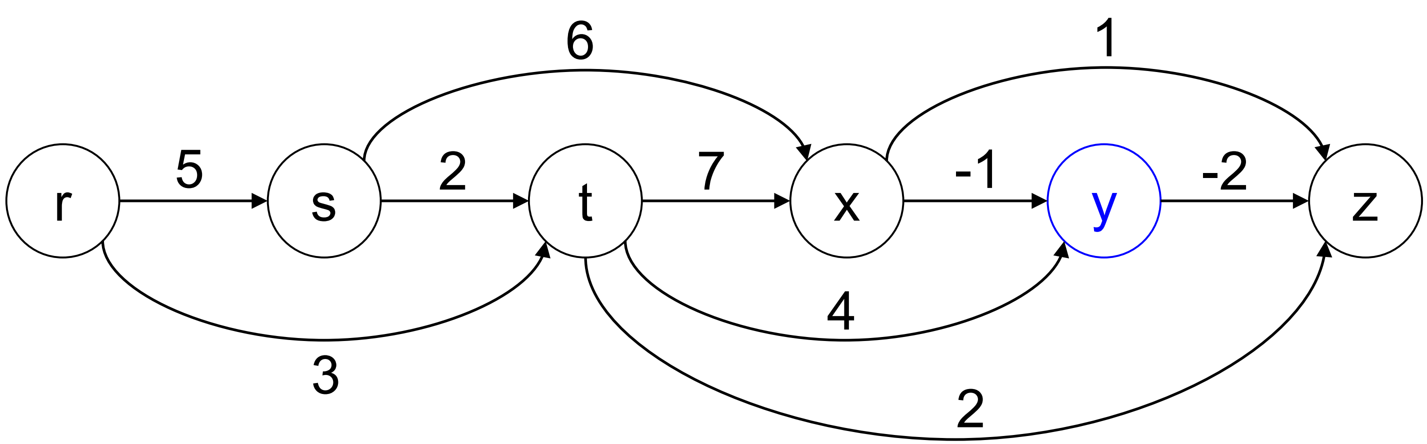
- Exemplo (a partir de s)
  - Arestas adjacentes a x



d(r)=∞ d(s)=0 d(t)=2 d(x)=6 d(y)=6→5 d(z)=4

Algoritmo baseado na ordenação topológica

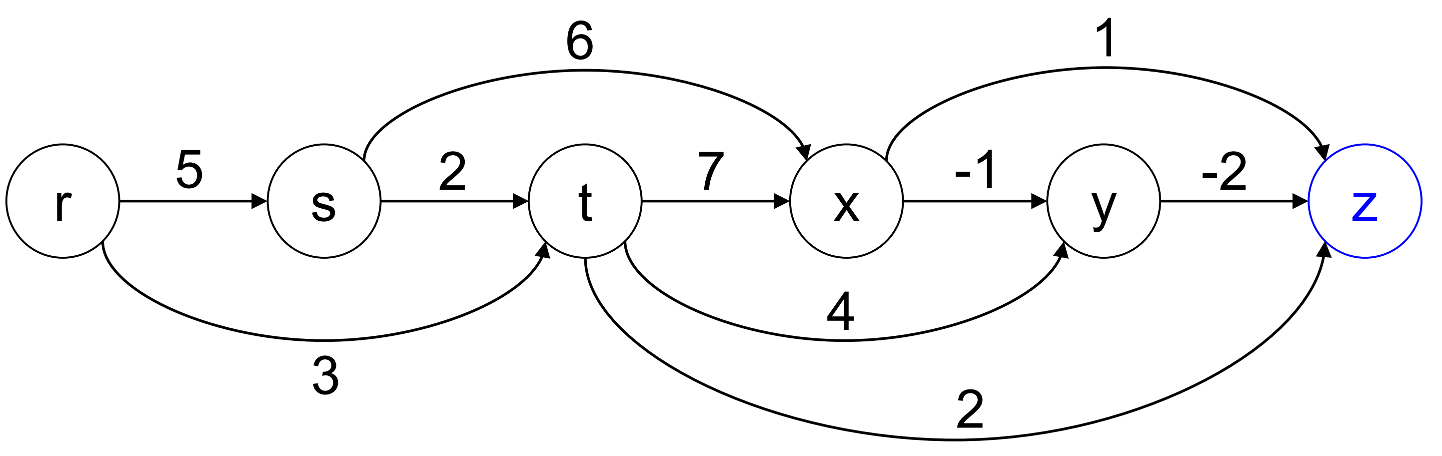
- Exemplo (a partir de s)
  - Arestas adjacentes a y



d(r)=∞ d(s)=0 d(t)=2 d(x)=6 d(y)=5 d(z)=4→3

Algoritmo baseado na ordenação topológica

- Exemplo (a partir de s)
  - Arestas adjacentes a z



d(r)=∞ d(s)=0 d(t)=2 d(x)=6 d(y)=5 d(z)=3

Algoritmo baseado na ordenação topológica

```
Caminho mínimo baseado na ordenação topológica(G, w, s)
início
    //ordenação topológica
    ordenar topologicamente o grafo (via busca em profundidade)
    //inicializa variáveis
    para cada vértice v faça
        d[v]=∞
        antecessor[v]=-1
    d[s]=0
    //faz relaxamento de arestas e determina caminhos mais curtos
    para cada vértice u tomado em sequência topológica faça
        para cada vértice v adjacente a u faça
            relax(u,v,w)
fim
```

Algoritmo baseado na ordenação topológica

- Complexidade de tempo: ?

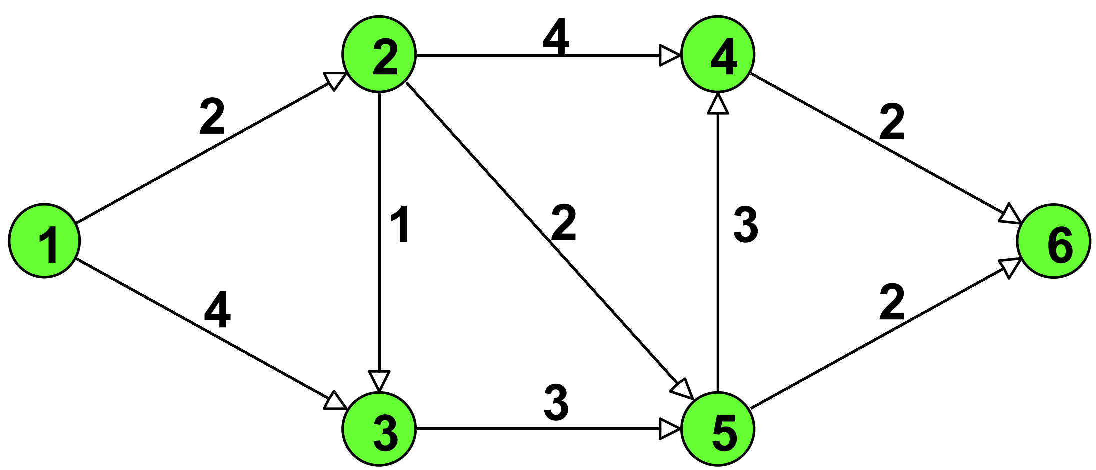
Algoritmo baseado na ordenação topológica

- Complexidade de tempo: $O(|V| + |A|)$
  - Por que?

Exercício

Exercício

- Calcule os caminhos mínimos para o grafo abaixo a partir do vértice 1 aplicando o algoritmo baseado na ordenação topológica



Caminhos Mais Curtos de **Todos os Pares**

- Suponha que um grafo orientado ponderado representa as possíveis rotas de uma companhia aérea conectando diversas cidades

- O objetivo é construir uma tabela com os menores caminhos entre todas as cidades

- Esse é um exemplo de problema que exige encontrar os **caminhos mais curtos para todos os pares de vértices**