Caminho mínimo parte I

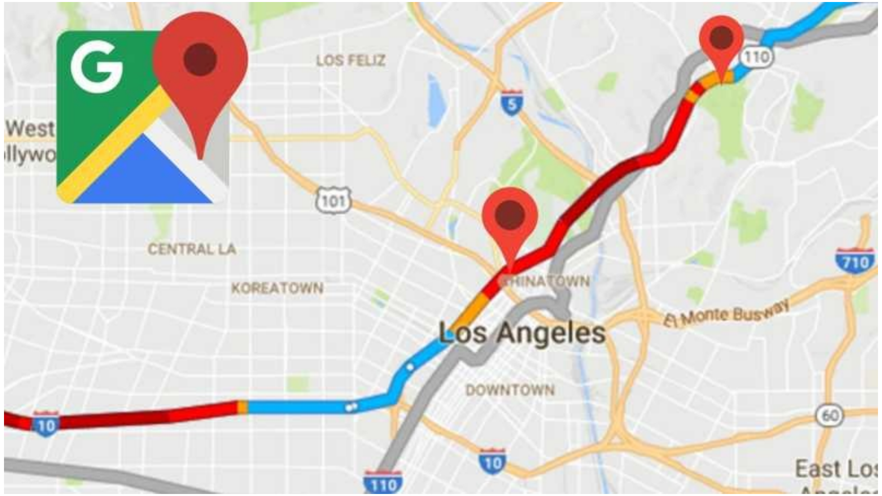
prof. Dr. Wesin Ribeiro

Neste capítulo

- Introdução
- Algoritmo de Bellman-Ford
- Caminhos mínimos de fonte única em grafos acíclicos dirigidos
- Algoritmo de Dijkstra
- Restrições e Diferenças
- Provas de propriedades
- Revisão

Introdução

Dando continuidade a disciplina de grafos, neste capítulo veremos como encontrar o caminho mínimo de forma eficiente. Imagine que um motorista deseja encontrar a rota mais curta possível do Rio de Janeiro a São Paulo. Dado um mapa rodoviário do Brasil no qual a distância entre cada par de interseções adjacentes esteja marcada, como ele pode determinar essa rota mais curta?



Qual é o caminho mais curto?

Em um problema de caminhos mínimos, temos um grafo dirigido ponderado G = (V, E), com função peso w que mapeia arestas para pesos de valores reais. O peso do caminho p é a soma dos pesos de suas arestas.

- Problema de caminho mínimo de fonte única
  - com um só destino
  - para um par
  - para todos os pares
- A subestrutura ótima do caminho mínimo
- Arestas de peso negativo
- Não pode conter ciclos de peso negativo

$$p = \langle v_0, v_1, \ldots, v_k \rangle$$

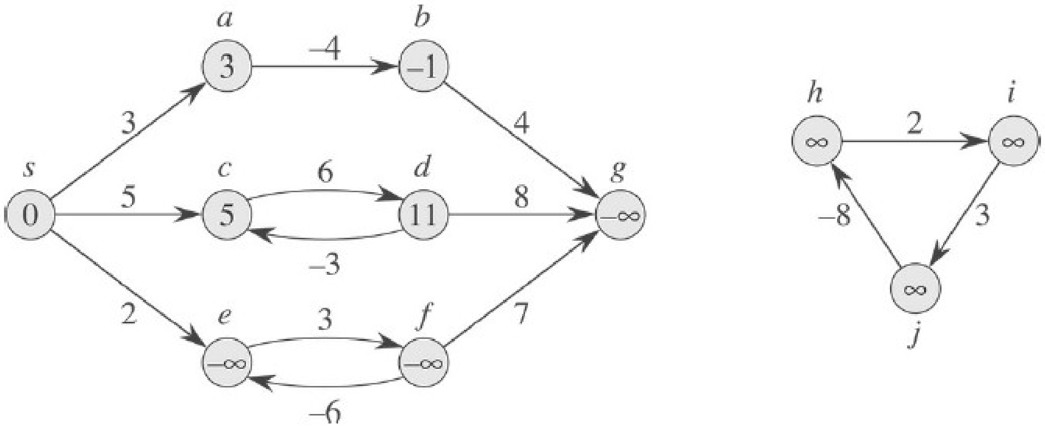
$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i) .$$

$$\delta(u, v) = \begin{cases} \min\{w(p) : u \overset{p}{\leadsto} v\} \\ \infty \end{cases}$$

Ciclos de peso negativo

Muitas vezes, desejamos calcular não apenas pesos de caminhos mínimos, mas também os vértices nos caminhos mínimos.

- Vértices (e,f) formam um ciclo de peso negativo
- H,i e j não podem ser alcançados por s



Relaxamento de arestas

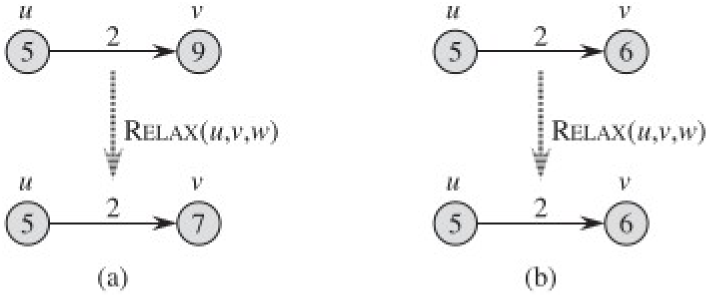
Muitas vezes, desejamos calcular não apenas pesos de caminhos mínimos, mas também os vértices nos caminhos mínimos.

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

$\quad \textbf{if } v.d > u.d + w(u, v)$

$\quad\quad v.d = u.d + w(u, v)$

$\quad\quad v.\pi = u$

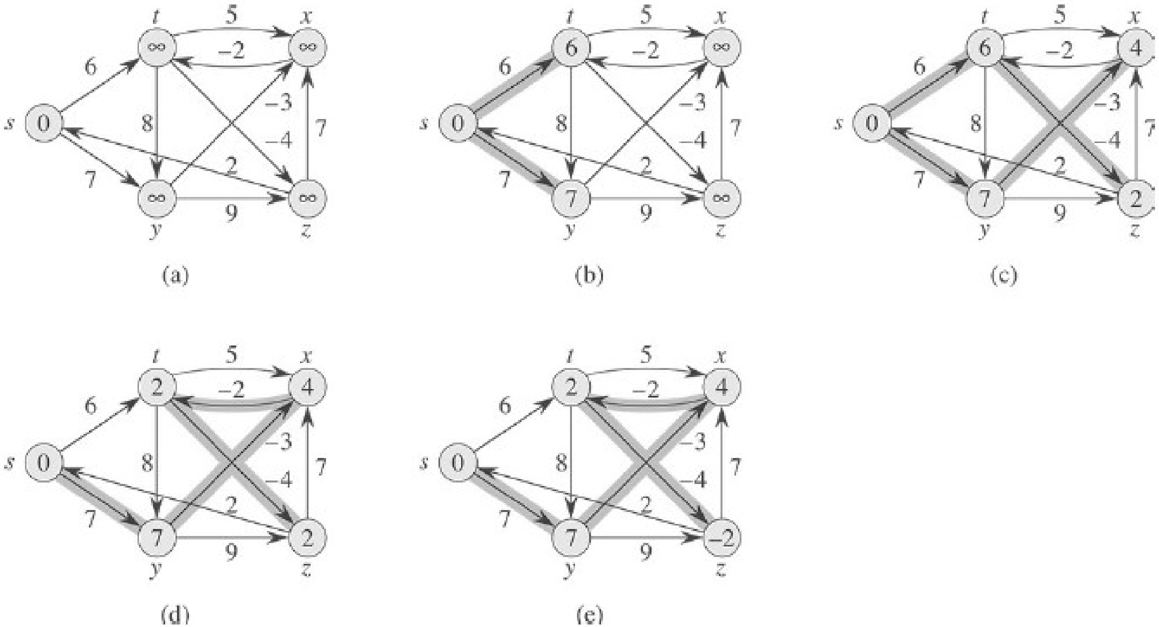


Algoritmo de Bellman-Ford

O algoritmo de Bellman–Ford resolve o problema de caminhos mínimos de fonte única no caso geral no qual os pesos das arestas podem ser negativos

O algoritmo retorna um valor booleano indicando se o grafo contém um caminho mínimo.

A idéia principal é relaxar as arestas diminuindo progressivamente o valor do vértice a cada iteração.



Algoritmo de Bellman-Ford

```
BELLMAN-FORD(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
for i = 1 to |G.V| − 1
    for each edge (u, v) ∈ G.E
        RELAX(u, v, w)
for each edge (u, v) ∈ G.E
    if v.d > u.d + w(u, v)
        return FALSE
return TRUE
```

```
INITIALIZE-SINGLE-SOURCE(G, s)
for each vertex v ∈ G.V
    v.d = ∞
    v.π = NIL
s.d = 0
```

$O(VE)$

Caminho mínimos em GAD

Caminhos mínimos são sempre bem definidos em um gad já que, mesmo que existam arestas de peso negativo, não deve existir nenhum ciclo de peso negativo.

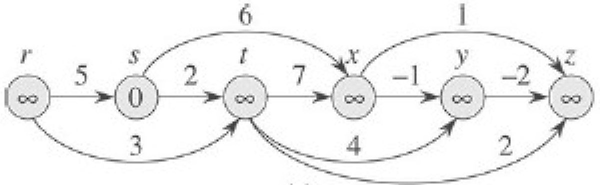
```
DAG-SHORTEST-PATHS (G, w, s)

ordenar topologicamente os vértices de G
INITIALIZE-SINGLE-SOURCE (G, s)
for cada vértice u tomado em ordem topológica
    for cada vértice v ∈ Adj[u]
        RELAX(u, v, w)
```

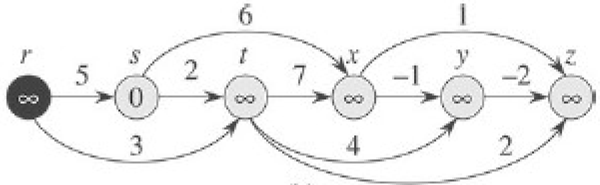
theta(V + E)

Caminho mínimos em GAD

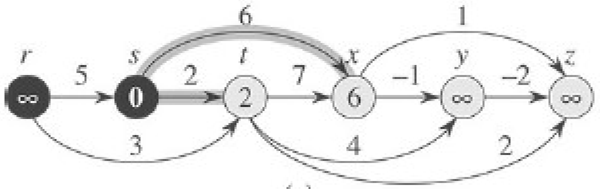
Caminhos mínimos são sempre bem definidos em um gad já que, mesmo que existam arestas de peso negativo, não deve existir nenhum ciclo de peso negativo.



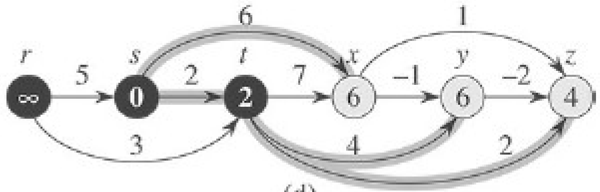
(a)



(b)



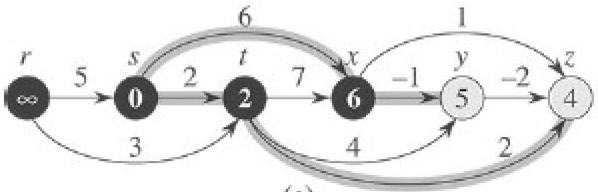
(c)



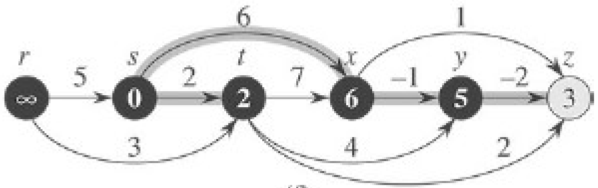
(d)

Caminho mínimos em GAD

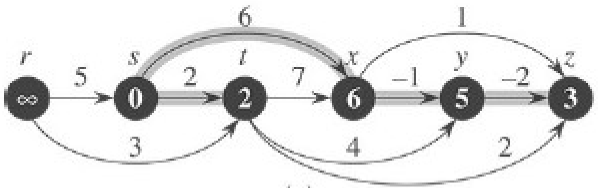
Se o gad contém um caminho do vértice u ao vértice v, então u precede v na ordem topológica



(e)



(f)

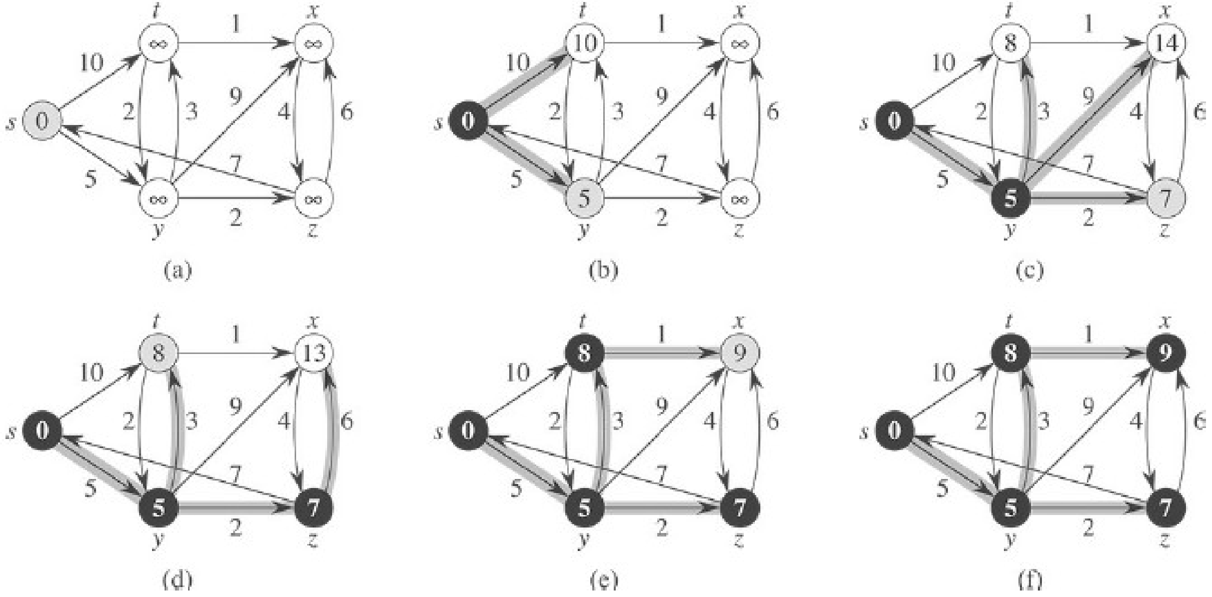


(g)

O algoritmo de Dijkstra

Resolve o problema de caminhos mínimos de fonte única em um grafo dirigido ponderado *G = (V, E) para o caso no qual todos os pesos de arestas são não negativos*.

Utiliza uma lista para armazenar os vértices ordenados e uma fila de prioridades Q na sua implementação.



O algoritmo de Dijkstra

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S = ∅
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

$O(V \log V + E)$

Exercícios

Execute o algoritmo de BellmanFord no grafo dirigido da Figura 24.4, usando o vértice z como fonte. Em cada passagem, relaxe arestas na mesma ordem da figura e mostre os valores de d e p após cada passagem.

Modifique o algoritmo de BellmanFord de modo que ele termine com v.d como – ∞ para todos os vértices v
para os quais existe um ciclo de peso negativo em algum caminho da fonte até v

Execute o algoritmo de Dijkstra para o grafo dirigido da Figura 24.2, primeiro usando o vértice s como fonte e
depois usando o vértice z como fonte. Mostre os valores de d e p e os vértices no conjunto S após cada iteração do laço while.

Implemente os algoritmos Dijkstra e BellmanFord em c++

Revisão

Atenção, chegou a hora da revisão.

Caminho mínimo é o caminho de menor custo que liga o vértice fonte a um vértice destino em um grafo dirigido e ponderado G.

A função peso calcula a soma de todos os pesos contidos em um determinado caminho do grafo G.

A rotina principal de um algoritmo que encontra todos os caminhos mínimos de um grafo é a rotina de relaxamento de arestas.

O algoritmo de BellmanFord detecta os caminhos mínimos de fonte única em um grafo dirigido, ponderado e que tenha arestas negativas.

O algoritmo de Dijkstra calcula os caminhos mínimos de fonte única usando uma fila de prioridades e atualiza as informações de pesos dos vertices e seus predecessores com o auxílio da rotina de relaxamento.