CI2613: Algoritmos y Estructuras III

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

Enero-Marzo 2015

Caminos de costo mínimo en grafos

Algoritmo de Bellman-Ford

Algoritmo de Bellman-Ford

El algoritmo de Bellman-Ford computa todas las distancias mínimas a todos los vértices de un grafo $G = (V, E)$ desde un vértice fuente s:

– Funciona para **pesos arbitrarios** $w : E \to \mathbb{R}$

– Si existe un ciclo de costo negativo alcanzable desde s, el algoritmo lo detecta

– Si no hay dicho ciclo, el algoritmo computa las distancias y un árbol de caminos de costo mínimo

Bellman-Ford: relajar cada arista del grafo $|V| - 1$ veces

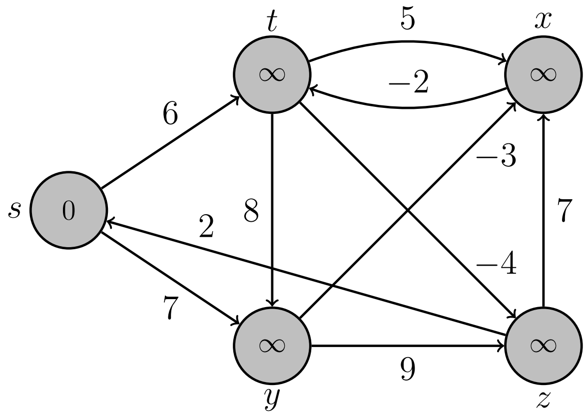
Bellman-Ford: Pseudocódigo

```
bool Bellman-Ford(G, w, s):
    Inicializar-vertice-fuente(G, s)
    for i = 1 to |V| - 1
        foreach (u,v) ∈ E
            Relajar(u, v, w)

    // detección de ciclo de costo negativo
    foreach (u,v) ∈ E
        if d[v] > d[u] + w(u,v)
            return false
    return true
```

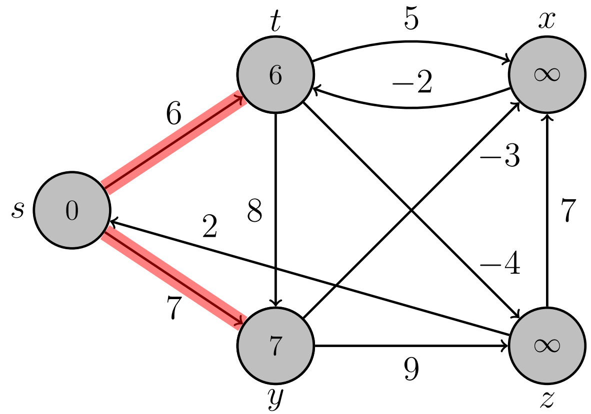
Bellman-Ford retorna **`true`** si y sólo si G no contiene un ciclo de costo negativo alcanzable desde s

Bellman-Ford: Ejemplo



$(t,x),(t,y),(t,z),(x,t),(y,x),(y,z),(z,x),(z,s),(s,t),(s,y)$

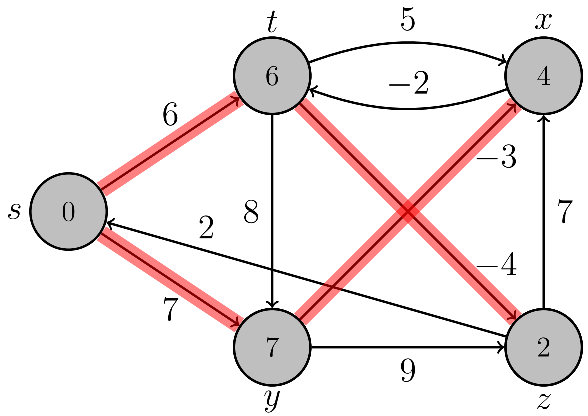
Bellman-Ford: Ejemplo



$(t,x),(t,y),(t,z),(x,t),(y,x),(y,z),(z,x),(z,s),(s,t),(s,y)$

Después de la primera iteración

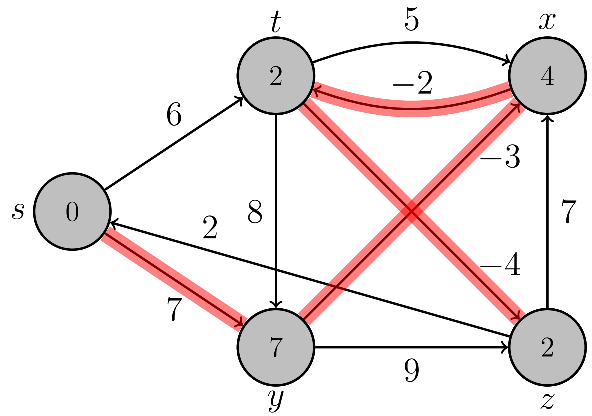
Bellman-Ford: Ejemplo



$(t,x),(t,y),(t,z),(x,t),(y,x),(y,z),(z,x),(z,s),(s,t),(s,y)$

Después de la segunda iteración

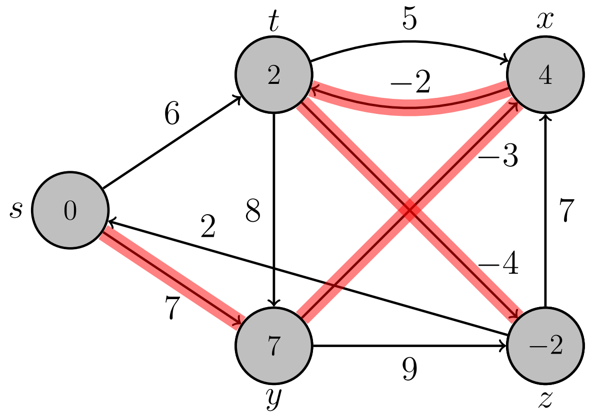
Bellman-Ford: Ejemplo



$(t,x),(t,y),(t,z),(x,t),(y,x),(y,z),(z,x),(z,s),(s,t),(s,y)$

Después de la tercera iteración

Bellman-Ford: Ejemplo



$(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)$

Después de la cuarta iteración

Bellman-Ford: Pseudocódigo

```
1  bool Bellman-Ford(G, w, s):
2      Inicializar-vertice-fuente(G, s)
3      for i = 1 to |V| - 1
4          foreach (u,v) ∈ E
5              Relajar(u, v, w)
6
7      // detección de ciclo de costo negativo
8      foreach (u,v) ∈ E
9          if d[v] > d[u] + w(u,v)
10             return false
11     return true
```


Bellman-Ford: Análisis

Claramente Bellman-Ford toma tiempo $\Theta(VE)$:

1. La inicialización toma tiempo $\Theta(V)$
2. Cada arista de relaja $\Theta(V)$ veces. Una relajación toma tiempo constante. El tiempo invertido en las relajaciones es $\Theta(VE)$
3. Determinar si existe un ciclo de costo negativo (segundo lazo) toma tiempo $O(E)$
4. Tiempo total: $\Theta(V) + \Theta(VE) + O(E) = \Theta(VE)$

Bellman-Ford: Correctitud

Lema

Sea $G = (V, E)$ un digrafo con vértice fuente s y pesos $w : E \to \mathbb{R}$. Asuma que G no tiene un ciclo de costo negativo alcanzable desde s. Luego de las $|V| - 1$ iteraciones del ciclo 3–5 en `Bellman-Ford`, $d[v] = \delta(s, v)$ para todos los vértices v.

Prueba: si v no es alcanzable desde s, por Invariante 2, $d[v] = \delta(s, v) = \infty$

Si v es alcanzable desde s, sea $p = (v_0, v_1, \ldots, v_k)$ un camino más corto de $v_0 = s$ a $v_k = v$ de **longitud** a lo sumo $|V| - 1$

Cada una de las $|V| - 1$ iteraciones del lazo relaja todos las aristas del grafo

En consecuencia, las aristas (v_0, v_1), (v_1, v_2), ..., (v_{k-1}, v_k) se relajan en orden. Por el Invariante 4, al finalizar el lazo, $d[v] = \delta(s, v)$ □

Bellman-Ford: Correctitud

Corolario

Sea $G = (V, E)$ un digrafo con vértice fuente s y pesos $w : E \to \mathbb{R}$. Asuma que G no tiene un ciclo de costo negativo alcanzable desde s. Para cada vértice v, existe un camino de s a v si y sólo si al terminar `Bellman-Ford` $d[v] < \infty$.

Bellman-Ford: Correctitud

Teorema

Considere una corrida de `Bellman-Ford` sobre un digrafo $G = (V, E)$ con vértice fuente s y pesos $w : E \to \mathbb{R}$.

*Si G no contiene un ciclo de costo negativo alcanzable desde s, el algoritmo retorna **`true`**, $d[v] = \delta(s, v)$ para todo $v \in V$ y el grafo de predecesores es un árbol de caminos más cortos.*

*Si G contiene un ciclo de costo negativo alcanzable desde s, el algoritmo retorna **`false`***

Bellman-Ford: Correctitud

Prueba:

Si G no tiene ciclo de costo negativo alcanzable desde s, el Lema implica que al terminar `Bellman-Ford`, $d[v] = \delta(s, v)$ para todo v

Por el Invariante 5, el árbol de predecesores es un árbol de caminos más cortos

Falta mostrar que `Bellman-Ford` retorna **`true`**

Al finalizar `Bellman-Ford`, para cada arista (u, v) tenemos:

$$d[v] \; = \; \delta(s, v) \; \leq \; \delta(s, u) + w(u, v) \; = \; d[u] + w(u, v)$$

Por lo tanto, ninguno de los tests dentro del lazo es cierto y `Bellman-Ford` retorna **`true`**

Bellman-Ford: Correctitud

Considere que G tiene un ciclo de costo negativo alcanzable desde s

Sea $c = (v_0, \ldots, v_k)$ dicho ciclo con $v_0 = v_k$ y $\sum_{i=1}^{k} w(v_{i-1}, v_i) < 0$

Asuma que `Bellman-Ford` retorna **`true`**. Entonces, para $i = 0, \ldots, k$, $d[v_i] \leq d[v_{i-1}] + w(v_{i-1}, v_i)$

Sumando las desigualdades:

$$\begin{aligned}
\textstyle\sum_{i=1}^{k} d[v_i] \; &\leq \; \textstyle\sum_{i=1}^{k} d[v_{i-1}] + w(v_{i-1}, v_i) \\
&= \; \textstyle\sum_{i=1}^{k} d[v_{i-1}] + \sum_{i=1}^{k} w(v_{i-1}, v_i) \\
&= \; \textstyle\sum_{i=1}^{k} d[v_i] + \sum_{i=1}^{k} w(v_{i-1}, v_i)
\end{aligned}$$

La segunda igualdad porque $v_0 = v_k$

Como $d[v_i] < \infty$ para $i = 0, \ldots, k$: $\quad 0 \leq \sum_{i=1}^{k} w(v_{i-1}, v_i)$ $\square$