CI2613: Algoritmos y Estructuras III

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

Enero-Marzo 2015

Caminos de costo mínimo sobre DAGs

Caminos de costo mínimo sobre DAGs

Bellman-Ford soluciona el problema de caminos de costo mínimo desde un vértice fuente para grafos generales en tiempo $\Theta(VE)$

En grafos acíclicos podemos mejorar el desempeño de forma significativa explotando un orden topológico del grafo

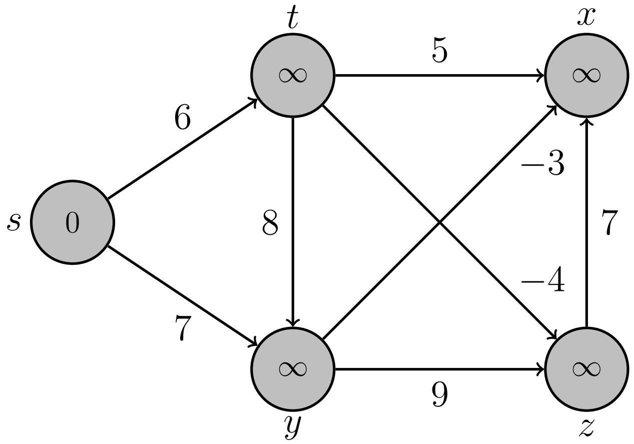
El algoritmo funciona para pesos arbitrarios

Caminos de costo mínimo sobre DAGs: Pseudocódigo

```
bool Caminos-Mas-Cortos-Sobre-DAG(G, w, s):
    Ordenar los vértices de G de forma topológica

    Inicializar-vertice-fuente(G, s)
    foreach Vertice u ∈ V en orden topológico
        foreach Vertice v ∈ adyacentes[u]
            Relajar(u, v, w)
```

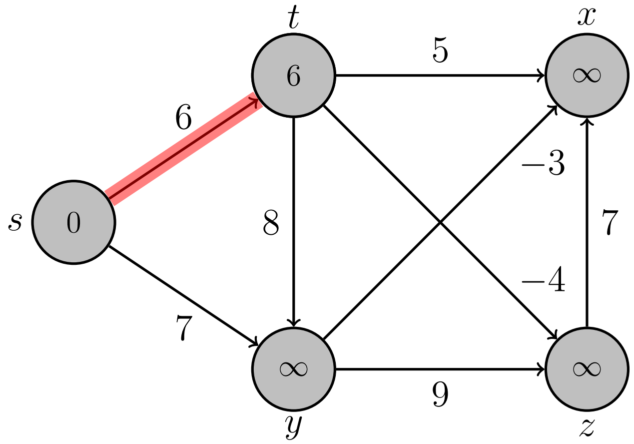

Caminos de costo mínimo sobre DAGs: Ejemplo



s, t, y, z, x

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

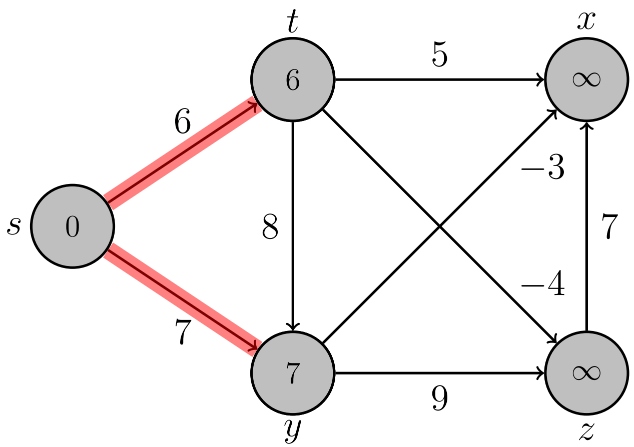
Caminos de costo mínimo sobre DAGs: Ejemplo



s, t, y, z, x

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

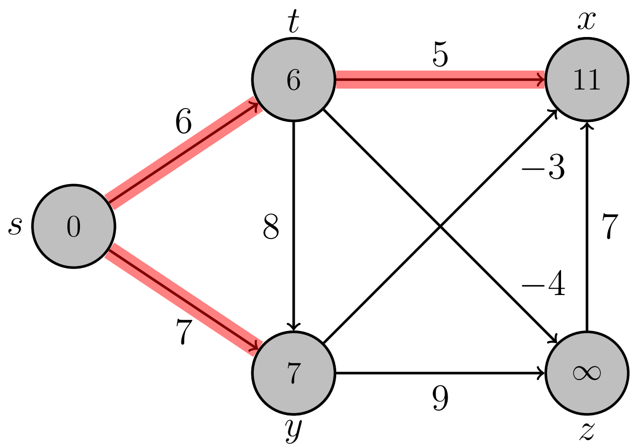
Caminos de costo mínimo sobre DAGs: Ejemplo



s, t, y, z, x

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

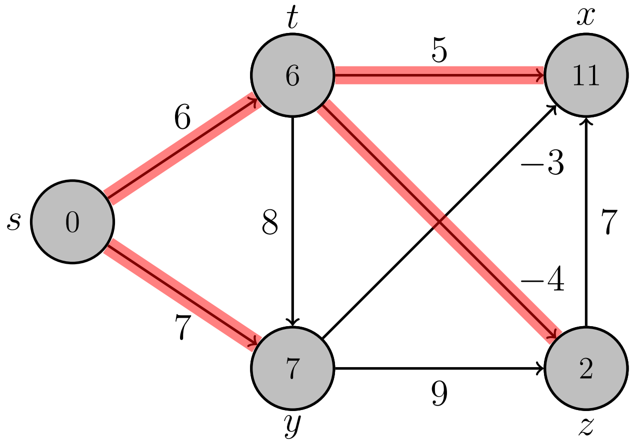
Caminos de costo mínimo sobre DAGs: Ejemplo



s, t, y, z, x

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

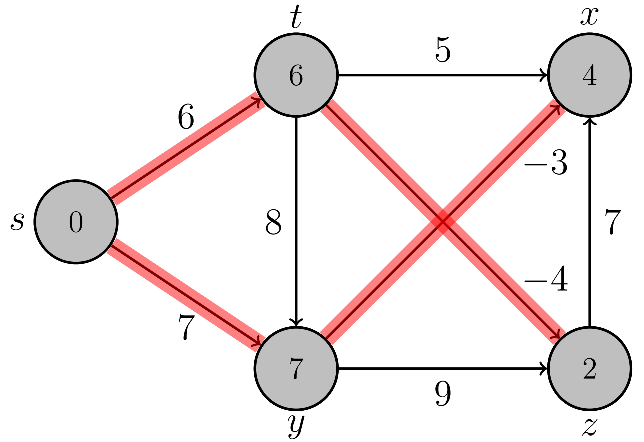
Caminos de costo mínimo sobre DAGs: Ejemplo



s, t, y, z, x

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

Caminos de costo mínimo sobre DAGs: Ejemplo



s, t, y, z, x

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

Caminos de costo mínimo sobre DAGs: Pseudocódigo

```
bool Caminos-Mas-Cortos-Sobre-DAG(G, w, s):
    Ordenar los vértices de G de forma topológica

    Inicializar-vertice-fuente(G, s)
    foreach Vertice u ∈ V en orden topológico
        foreach Vertice v ∈ adyacentes[u]
            Relajar(u, v, w)
```


Caminos de costo mínimo sobre DAGs: Análisis

1. El ordenamiento topológico toma tiempo $\Theta(V + E)$
2. La inicialización toma tiempo $\Theta(V)$
3. Las relajaciones toman tiempo $\Theta(E)$
4. Tiempo total: $\Theta(V + E) + \Theta(V) + \Theta(E) = \Theta(V + E)$

Caminos de costo mínimo sobre DAGs: Correctitud

Teorema

Sea $G = (V, E)$ un digrafo con vértice fuente s, sin ciclos y con pesos $w : E \to \mathbb{R}$. Al terminar `Caminos-Mas-Cortos-Sobre-DAG`*, $d[v] = \delta(s, v)$ para todo vértice v y el grafo de predecesores es un árbol de caminos más cortos*

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 $v_k = v$

Por ordenamiento, los vértices en p son procesados en orden $v_0, \ldots, v_k$

Las aristas de p son relajadas en orden (v_0, v_1), (v_1, v_2), $\ldots$, (v_{k-1}, v_k)

Por Invariante 4, al finalizar el algoritmo, $d[v] = \delta(s, v)$. Por Invariante 5, al finalizar, el grafo de predecesores es un árbol de caminos más cortos □

Aplicación: Control de proyectos

El algoritmo anterior se utliza en el área de control de proyectos para detectar el **camino crítico** en un diagrama PERT

PERT (Project Evaluation and Review Technique) es una técnica desarrollada a mediados de los 50s por la Armada de los E.E.U.U. en el contexto del desarrollo del **submarino nuclear Polaris**

Un diagrama PERT es un DAG $G = (V, E)$ donde los vértices denotan **hitos en el proyecto** (momentos donde se alcanza algún objetivo) y las aristas denotan **actividades**

Los pesos $w : E \to \mathbb{R}$ denotan los **tiempos esperados de duración** de las actividades

Aplicación: Control de proyectos

El grafo PERT contiene dos vértices designados s y t que denotan el comienzo y el final del proyecto

Un camino de costo mínimo de s a t representa una **secuencia crítica de actividades** tales que cualquier retraso en alguna de ellas, retrasaría la culminación del proyecto

El costo $\delta(s, t)$ de s a t en el grafo PERT es una cota inferior al tiempo de finalización del proyecto