# Caminos de costo mínimo sobre DAGs

Guillermo Palma

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

## Sobre los caminos de costo mínimos sobre DAGs

- El algoritmo encuentra los caminos de costo mínimo desde un vértice fuente fijo $s$, en un digrafo acíclico (DAG) $G = (V, E)$.
- Se tiene como entrada un digrafo $G$, una función de costo sobre los lados $w : E \to \mathbb{R}$ y un vértice fuente $s$.
- Relajando los lados de $G$ usando el orden topológico de los vértices, se puede obtener los caminos de costo mínimo en $\Theta(|V| + |E|)$.
- Como $G$ es un DAG, entonces no hay ciclos negativo.
- Por lo tanto, el problema de encontrar los caminos de costo mínimo en un DAG $G$ desde un vértice $s$ un fuente fijo está bien definido.
- Primero se obtiene el orden topológico de los vértices.
- Si hay un camino hasta $v$ que pase por $u$, entonces $u$ precede a $v$ en el orden topológico.
- Se hace un recorrido sobre los vértices usando el orden topológico.
- Para cada uno de los vértices $v \in V$, se relajan los lados que tienen como vértice de salida a $v$.

## Algoritmo de caminos de costo mínimo sobre DAGs

**Procedimiento** DAG-Caminos-Costo-Minimo($G = (V, E)$, $w$, $s$)

```
1 inicio
2     Hacer un ordenamiento topológico de los vértices de G ;
3     Inicializar-fuente-fija(G, s);
4     para cada (u ∈ V en el orden del ordenamiento topológico de G)
       hacer
5         para cada v ∈ G.adyacentes[u] hacer
6             Relajacion(u, v, w)
```



## Análisis del algoritmo de caminos de costo mínimo sobre DAGs

- El ordenamiento topológico es $\Theta(|V| + |E|)$.
- Inicializar-fuente-fija es $\Theta(|V|)$.
- Las líneas 4-6 indican que se ejecuta Relajacion (que es $O(1)$) a todos los lados del grafo una vez, esto es $O(|E|)$.
- Por lo tanto, el algoritmo de caminos de costo mínimo sobre DAGs es $\Theta(|V| + |E|)$.

## Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs

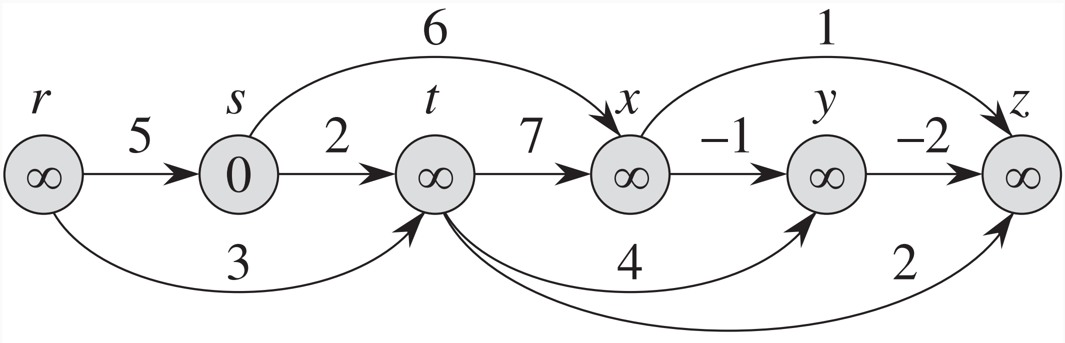


**Figura 1:** Estado del digrafo $G$ después del ordenamiento topológico e Inicializar-fuente-fija($G$, $s$). El vértice fuente es $s$. Fuente [1].



## Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs, cont.

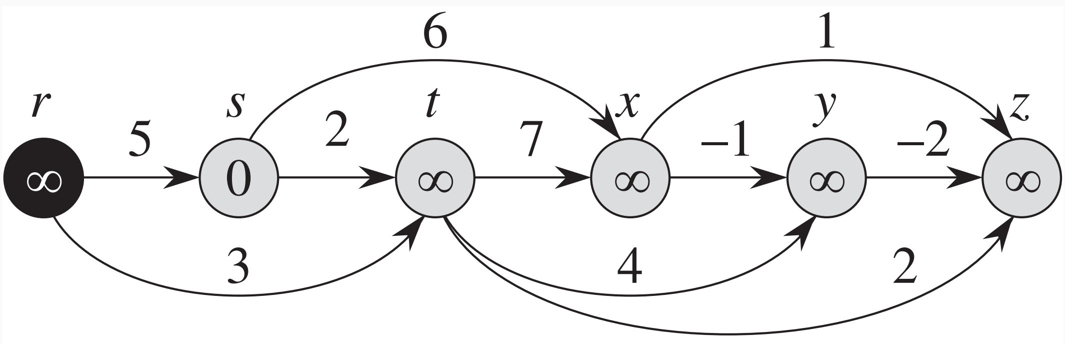


**Figura 2:** Resultado de los ciclos **para** en las líneas 4-6 aplicadas cuando el vértice es $r$. Fuente: [1].

# Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs, cont.

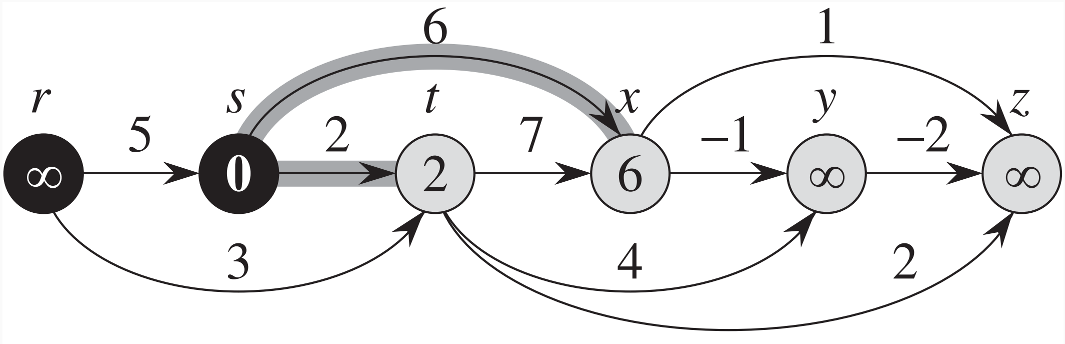


**Figura 3:** Resultado de los ciclos **para** en las líneas 4-6 aplicadas cuando el vértice es $s$. Fuente: [1].



# Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs, cont.

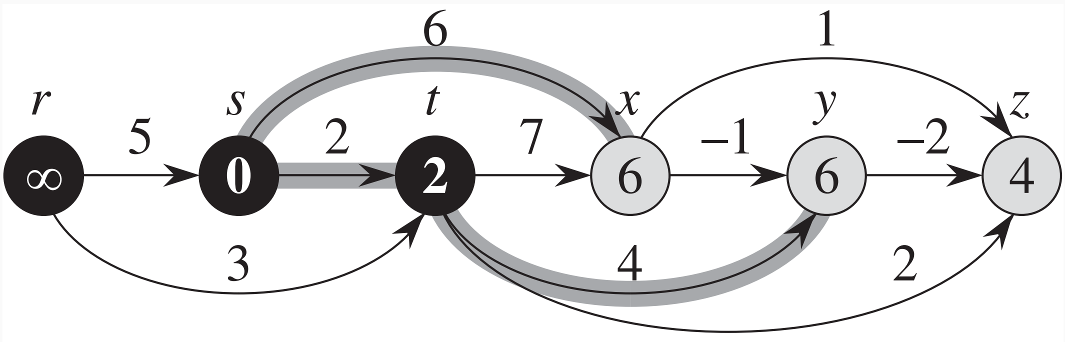


**Figura 4:** Resultado de los ciclos **para** en las líneas 4-6 aplicadas cuando el vértice es $t$. Fuente: [1].

## Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs, cont.

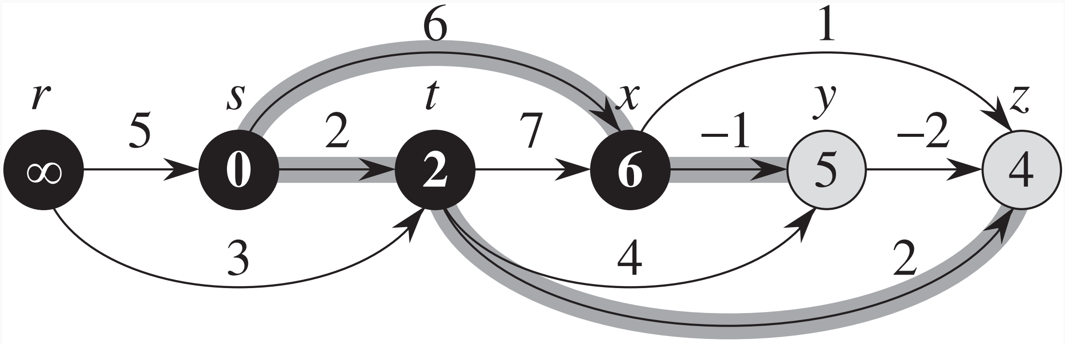


**Figura 5:** Resultado de los ciclos **para** en las líneas 4-6 aplicadas cuando el vértice es $x$. Fuente: [1].



## Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs, cont.

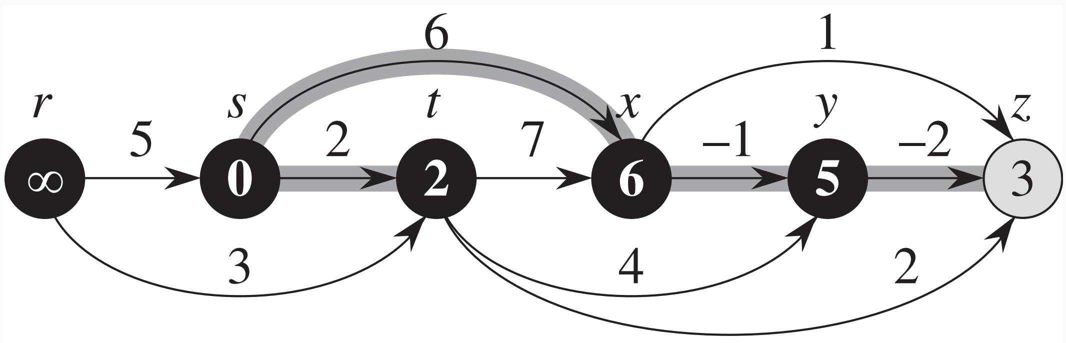


**Figura 6:** Resultado de los ciclos **para** en las líneas 4-6 aplicadas cuando el vértice es $y$. Fuente: [1].

## Ejemplo del algoritmo de caminos de costo mínimo sobre DAGs, cont.

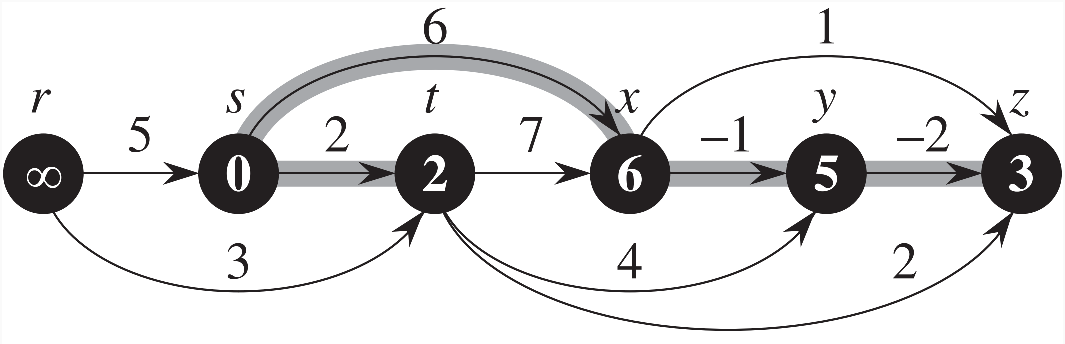


**Figura 7:** Resultado de los ciclos **para** en las líneas 4-6 aplicadas cuando el vértice es $z$. Fuente: [1].



## Correctitud de DAG-Caminos-Costo-Minimo

**Teorema correctitud de DAG-Caminos-Costo-Minimo**

Sea un digrafo $G = (V, E)$ acíclico, con una función de costo $w : E \to \mathbb{R}$ que asigna un costo real a cada uno de los lados del grafo, y dado un vértice fuente fijo $s \in V$. Entonces, cuando el procedimiento DAG-Caminos-Costo-Minimo termina se tiene que para todos los vértices $v \in V$ se cumple que $v.d = \delta(s, v)$, y el subgrafo predecesor $G_{pred}$ es un árbol de caminos de costo mínimo.

## Correctitud de DAG-Caminos-Costo-Minimo, cont.

**Prueba**

- Si $v$ no es alcanzable desde $s$, entonces $v.d = \infty$ por *propiedad de ningún camino*.
- Si $v$ es alcanzable desde $s$, entonces existe un camino de costo mínimo $p = \langle v_0, v_1, \ldots, v_k \rangle$, donde $v_0 = s$ y $v_k = v$.
- Como los vértices están en orden, entonces la relajación de los lados de $p$ se ejecuta en el orden $(v_0, v_1), (v_1, v_2) \ldots, (v_{k-1}, v_k)$.
- Por *propiedad de la relajación del camino*, entonces $v_i.d = \delta(s, v_i)$ para $i = 0, 1, \ldots, k$ cuando el procedimiento termina.
- Por *propiedad subgrafo-predecesor* se obtiene un árbol $G_{pred}$ de caminos de costo mínimo con raíz en $s$. $\square$



## Aplicación de DAG-Caminos-Costo-Minimo

- El procedimiento se puede aplicar para encontrar **caminos críticos** de un gráfico PERT.
- PERT es acrónimo de "*Program Evaluation and Review Technique*"
- Los lados son trabajos a ser ejecutados.
- Los costos de los lados son el tiempo que toma ejecutar un trabajo.
- Si se tienen los arcos $(u, v)$ y $(v, x)$, entonces el trabajo $(u, v)$ debe ser terminado antes que el trabajo $(v, x)$.
- Un camino del DAG es una secuencia de trabajos que deben ser ejecutados en ese orden.
- Un **camino crítico** es el camino más largo a través de DAG.
- Un **camino crítico** corresponde al *tiempo más largo en ejecutar* cualquier secuencia de trabajos.
- En consecuencia, las sumas de los costos del **camino crítico** da una cota inferior del tiempo que tomaría ejecutar todos los trabajos.

## Obtención del camino crítico

**Opción 1:** Negando los costos del digrafo de entrada y luego ejecutando DAG-Caminos-Costo-Minimo.

**Opción 2:** Ejecutando DAG-Caminos-Costo-Minimo, con la modificación de reemplazar $\infty$ por $-\infty$ en la línea 3 Inicializar-fuente-fija, y luego cambiar "$>$"por "$<$", en el procedimiento Relajacion en la línea 6.



## Referencias

[1] T. Cormen, C. Leirserson, R. Rivest, and C. Stein.
**Introduction to Algorithms.**
McGraw Hill, 3ra edition, 2009.