# UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN

## Taller de programación II - IN1071C

## Árbol Recubridor de Peso Mínimo

Diego Maldonado

Facultad de Ingeniería
Departamento de Ingeniería Informática

9 de noviembre de 2022

Contenido

## Definición (Árbol)

Un grafo $G = (V, E)$ es árbol si todos no tiene ciclosy es conexo.

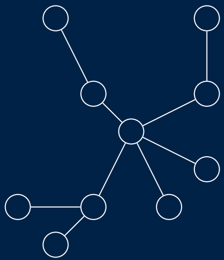
Si además al árbol se le define un vértice especial llamado raíz, , entonces se puede definir una distancia a la raíz como el número de aristas necesarios para llegar a la raíz.
Los vértices de grado 1 se llaman hojas. La mayor distancia entre la raíz y las hoja se llama profundidad.

## Definición (Árbol)

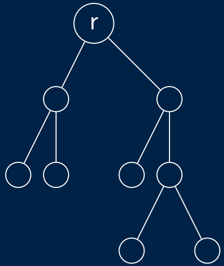
Un grafo $G = (V, E)$ es árbol si todos no tiene ciclosy es conexo.

Si además al árbol se le define un vértice especial llamado raíz, entonces se puede definir una distancia a la raíz como el número de aristas necesarios para llegar a la raíz.
Los vértices de grado 1 se llaman hojas. La mayor distancia entre la raíz y las hoja se llama profundidad.

## Ejemplo



Árbol




Árbol binario

## Definición

Dado un grafo $G = (V, E)$, un árbol recubridor (o de expansión) $T$ es un subgrafo de $G$ que satisface:

- $V(T) = V$, tiene todos lo vértices
- es un árbol.

## Definición

Dado un grafo $G = (V, E)$, una función de pesos $w : E \to \mathbb{R}$ y un árbol recubridor del grafo $T$, definimos:

- El peso del árbol $T$ como $w(T) = \sum_{e \in E(T)} w(e)$,
- El peso del árbol recubridor de peso mínimo de $G$ como $\delta(T)$.

## Definición

Problema: Árbol Recubridor de Peso Mínimo (ARPM):
¿Cuál es el árbol de de menor peso que contiene todos los vértices de G?

## Definición

Dado un grafo $G = (V, E)$, definimos:

- La función de pesos $w : E \to \mathbb{R}^+$
- El peso de un camino $v_1 ... v_n$ como $w(v_1 ... v_n) = \sum_{i=1}^{n-1} w(v_i v_{i+1})$
- El peso del camino más corto entre $u$ y $v$ como $\delta(u, v)$.

## Definición

Problema del camino más corto desde $s$ en $G$:
¿Cuáles son los caminos más cortos desde $s$ a cualquier otro vértice?

## Algoritmo de Dijkstra

**Entrada**:

- $G(V, E)$ grafo conexo,
- $s$ fuente,
- $w$ pesos positivos.

---

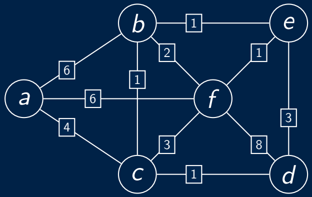
**Salida**: $D[n]$: distancia más corta a $s$ y $P[n]$: predecesor de cada vértice.

---

**Para** $v \in V \setminus s$ **hacer**
  $D[v] \leftarrow \infty$
  $P[v] \leftarrow \perp$
$D[s] \leftarrow 0$; $P[s] \leftarrow s$
$V' \leftarrow V$; $T \leftarrow \varnothing$
**Mientras** $V' \neq \varnothing$ **hacer**
  **Buscar** $v \in V'$ con mínimo valor de $D$.
  **Para** $u \in N_{(+)}(v) \cap V'$ **hacer**
    **Si** $D[v] + w(v, u) < D[u]$ **entonces**
      $D[u] \leftarrow D[v] + w(u, v)$
      $P[u] \leftarrow v$
  $V' \leftarrow V' \setminus v$
  $T \leftarrow T \cup v$

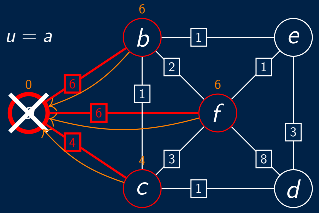
## Ejemplo

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos:



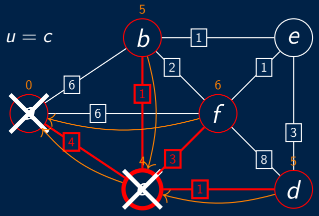
## Ejemplo (solución)

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos:



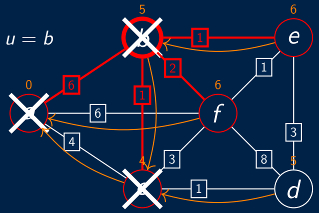
## Ejemplo (solución)

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos:



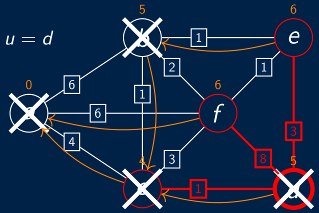
## Ejemplo (solución)

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos:



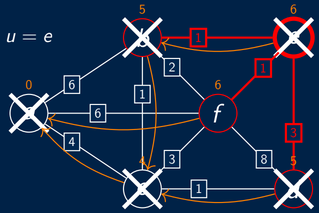
## Ejemplo (solución)

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos:



## Ejemplo (solución)

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos:



## Ejemplo (solución)

Encuentre la distancia mínima desde $s$ al resto de los vértices en el siguiente grafo con pesos: