

Tema 6.1: Árbol y Arborescencia de Unión de Valor Óptimo

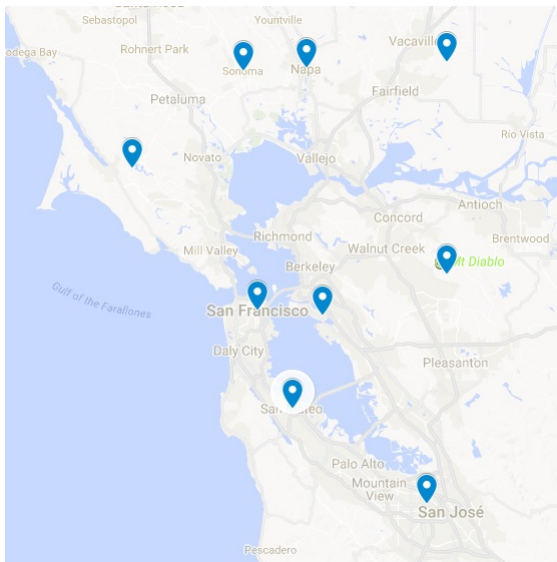
Modelización Matemática
Máster en Ingeniería de Caminos, Canales y Puertos

Ignacio Montes
Departamento de Estadística e I.O. y D.M.

¿Qué vamos a hacer en este tema?

- Definir árboles y bosques.
- Árboles de unión.
- Árboles de unión de valor mínimo (AUVM):
 - Algoritmo de Kruskal.
 - Algoritmo de Prim.
- Arborescencias.

Ejemplo: Conexión entre la bahía de San Francisco



Fuente: Google Maps.

1 Árboles

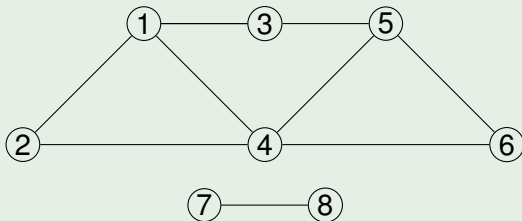
2 Arbol de unión de valor mínimo

- Planteamiento del problema
- Algoritmo de Kruskal
- Algoritmo de Prim

3 Arborescencias

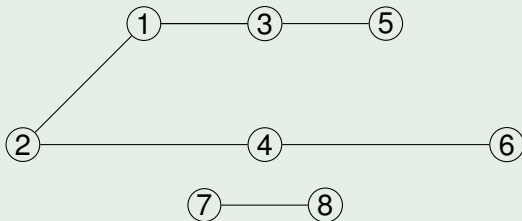
- Un **bosque** es un grafo $G = (V, A)$ no orientado y acíclico.

Ejemplo (No es un bosque: hay ciclos)



- Un **bosque** es un grafo $G = (V, A)$ no orientado y acíclico.

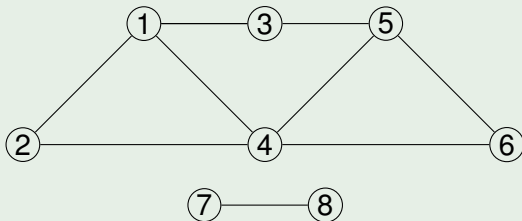
Ejemplo (Es un bosque: acíclico)



Árboles y bosques

- Un **bosque** es un grafo $G = (V, A)$ no orientado y acíclico.
- Un **árbol** es un grafo $G = (V, A)$ no orientado, conexo y acíclico.

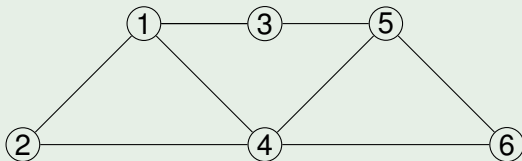
Ejemplo (No es un árbol: hay ciclos y no es conexo)



Árboles y bosques

- Un **bosque** es un grafo $G = (V, A)$ no orientado y acíclico.
- Un **árbol** es un grafo $G = (V, A)$ no orientado, conexo y acíclico.

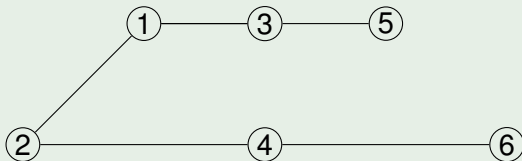
Ejemplo (No es un árbol: hay ciclos)



Árboles y bosques

- Un **bosque** es un grafo $G = (V, A)$ no orientado y acíclico.
- Un **árbol** es un grafo $G = (V, A)$ no orientado, conexo y acíclico.
- Un bosque está formado por varios árboles.

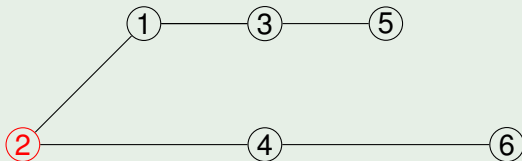
Ejemplo (Es un árbol: conexo y acíclico)



Árboles y bosques

- Un **bosque** es un grafo $G = (V, A)$ no orientado y acíclico.
- Un **árbol** es un grafo $G = (V, A)$ no orientado, conexo y acíclico.
- Un bosque está formado por varios árboles.
- La **raíz** de un árbol es uno de sus vértices.

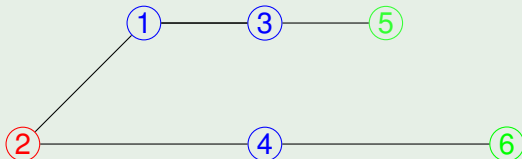
Ejemplo (Es un árbol con raíz en el vértice 2)



Raíz, nudo, hojas

- **Raíz:** vértice “inicial”.
- **Nudos:** vértices “intermedios” que cumplen $g(v) > 1$.
- **Hojas:** vértices “finales” que cumplen $g(v) = 1$.
- **Nivel de un vértice:** longitud mínima de una cadena desde la raíz al vértice.

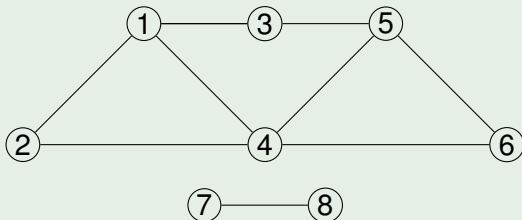
Ejemplo



Árbol de unión

- Dado un grafo $G = (V, A)$ no orientado, un **bosque de unión** es un bosque $G' = (V, A')$, donde $A' \subseteq A$.
- El bosque de unión siempre existe (por ejemplo, con $A' = \emptyset$).

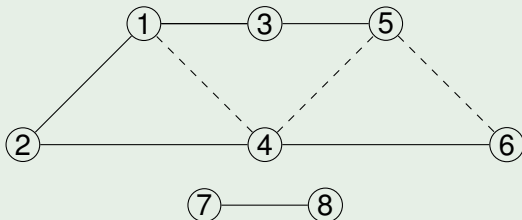
Ejemplo (Grafo no orientado)



Árbol de unión

- Dado un grafo $G = (V, A)$ no orientado, un **bosque de unión** es un bosque $G' = (V, A')$, donde $A' \subseteq A$.
- El bosque de unión siempre existe (por ejemplo, con $A' = \emptyset$).

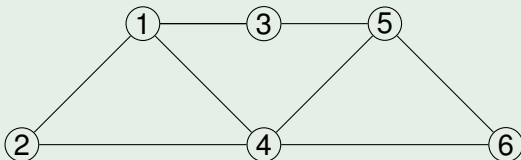
Ejemplo (Bosque de unión (elimino aristas discontinuas))



Árbol de unión

- Dado un grafo $G = (V, A)$ no orientado, un **bosque de unión** es un bosque $G' = (V, A')$, donde $A' \subseteq A$.
- El bosque de unión siempre existe (por ejemplo, con $A = \emptyset$).
- Dado un grafo $G = (V, A)$ no orientado, un **árbol de unión** es un árbol $G' = (V, A')$, donde $A' \subseteq A$.
- El árbol de unión existe siempre que el grafo G sea conexo.

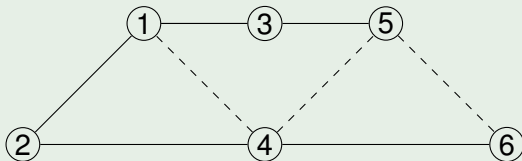
Ejemplo (Grafo no orientado y conexo)



Árbol de unión

- Dado un grafo $G = (V, A)$ no orientado, un **bosque de unión** es un bosque $G' = (V, A')$, donde $A' \subseteq A$.
- El bosque de unión siempre existe (por ejemplo, con $A = \emptyset$).
- Dado un grafo $G = (V, A)$ no orientado, un **árbol de unión** es un árbol $G' = (V, A')$, donde $A' \subseteq A$.
- El árbol de unión existe siempre que el grafo G sea conexo.

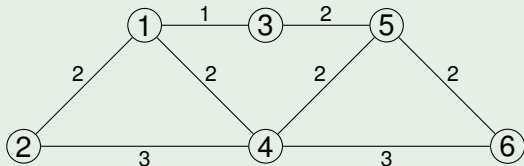
Ejemplo (Árbol de unión (elimino aristas discontinuas))



Descripción del problema

- Partimos de una red no orientada: $R = (V, A, p)$.
- Buscamos una red no orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es un árbol de unión del grafo no orientado $G = (V, A)$, y de manera que se minimiza $\sum_{e \in A'} p_e$.

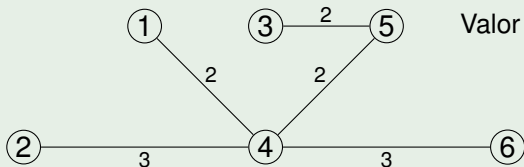
Ejemplo (Red no orientada)



Descripción del problema

- Partimos de una red no orientada: $R = (V, A, p)$.
- Buscamos una red no orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es un árbol de unión del grafo no orientado $G = (V, A)$, y de manera que se minimiza $\sum_{e \in A'} p_e$.

Ejemplo (Árbol de unión, ejemplo 1)

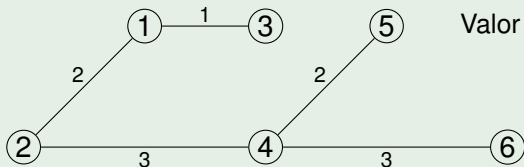


Valor del grafo: 12

Descripción del problema

- Partimos de una red no orientada: $R = (V, A, p)$.
- Buscamos una red no orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es un árbol de unión del grafo no orientado $G = (V, A)$, y de manera que se minimiza $\sum_{e \in A'} p_e$.

Ejemplo (Árbol de unión, ejemplo 2)



Valor del grafo: 11

- **Algoritmo de Kruskal:** añade a las aristas de menor valor que no forman ciclos.
- **Algoritmo de Prim:** parte de un vértice dado y va añadiendo vértices y aristas que minimicen el valor y no formen ciclos.

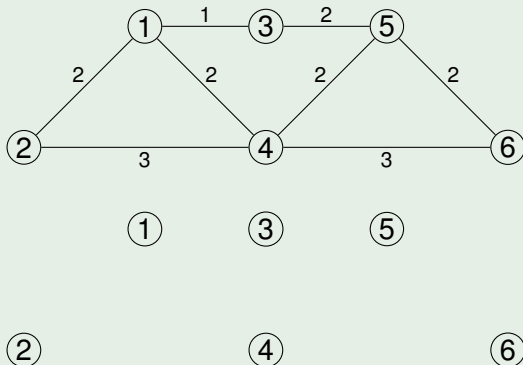
Algoritmo de Kruskal

Partimos de $R = (A, V, p)$ donde $G = (V, A)$ es un grafo no orientado.

- 1 Defino $A' = \emptyset$, $\mathcal{A} = A$.
- 2 Sea e^* una arista de manera que $p_{e^*} = \min\{p_e : e \in \mathcal{A}\}$.
 - 1 Si e^* forma un ciclo con las aristas en A' , elimino e^* de $\mathcal{A}$.
 - 2 Si e^* no forma un ciclo con las aristas de A' , añado e^* a A' y lo elimino de $\mathcal{A}$.
- 3 Iterar el paso 2 hasta que $\mathcal{A} = \emptyset$.

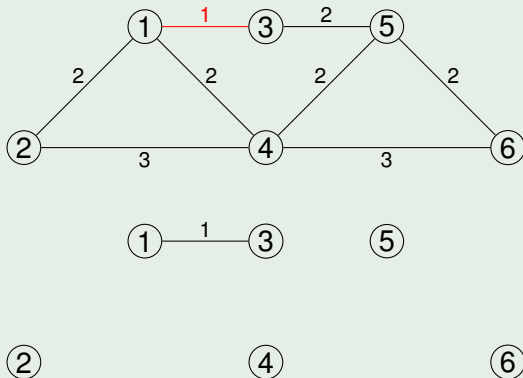
Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)

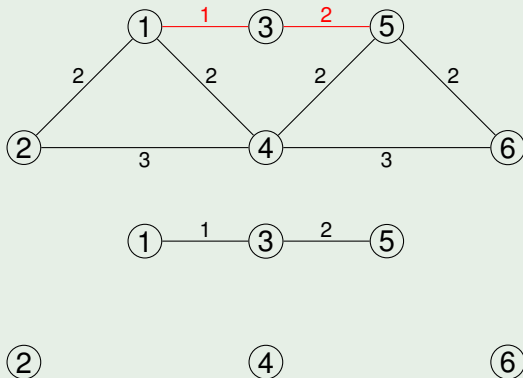


Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)

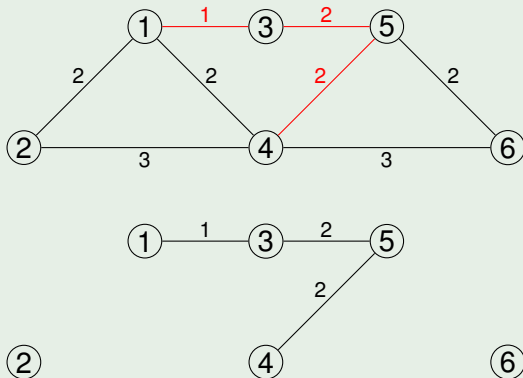


Ejemplo (AUVM mediante Kruskal)



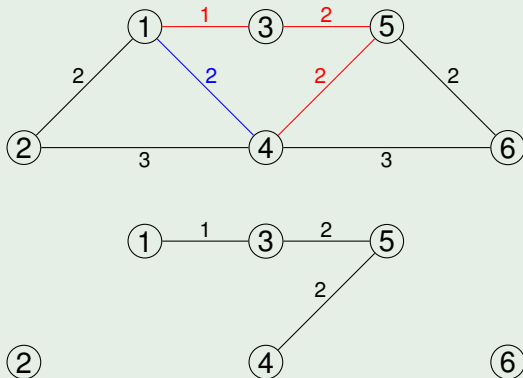
Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)



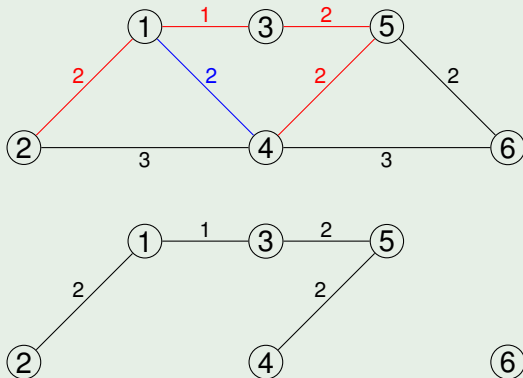
Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)



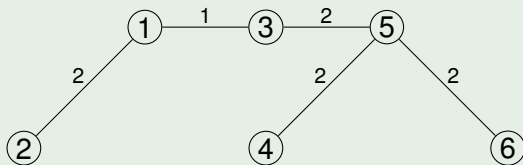
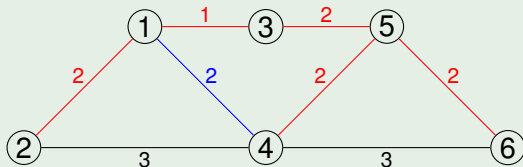
Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)



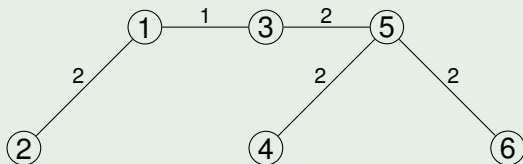
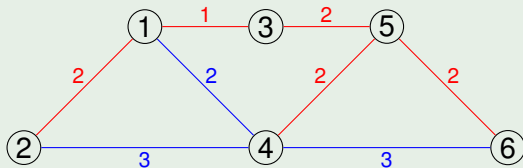
Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)

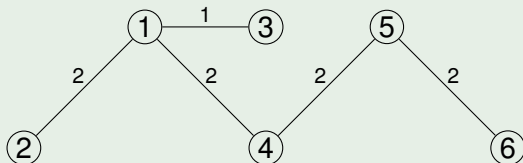
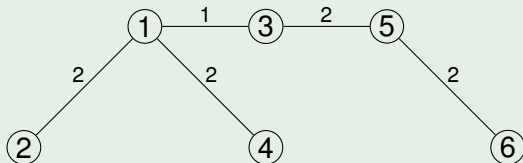


Algoritmo de Kruskal: ejemplo

Ejemplo (AUVM mediante Kruskal)



Ejemplo (Para este ejemplo hay 3 AUVM distintos)



Algoritmo de Prim

Partimos de $R = (A, V, p)$ donde $G = (V, A)$ es un grafo no orientado, y seleccionamos $v \in V$.

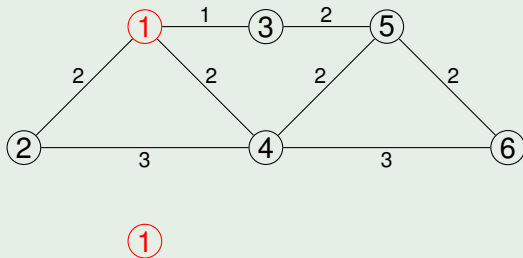
- 1 Defino $V' = \{v\}$ y $A' = \emptyset$.
- 2 Sea $e^* = (x^*, y^*) \in A$ de manera que:

$$p_{e^*} = \min\{p_e \mid e = (x, y) \text{ con } x \in V', y \notin V'\}.$$

- 3 Añado y^* a V' y e^* a A' .
- 4 Volver al paso 2 hasta que $V = V'$.

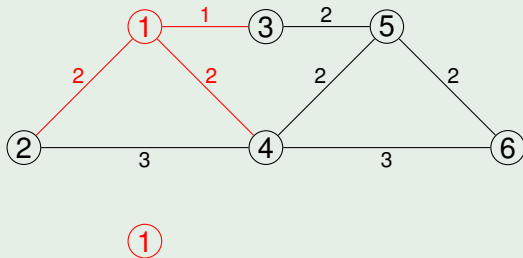
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



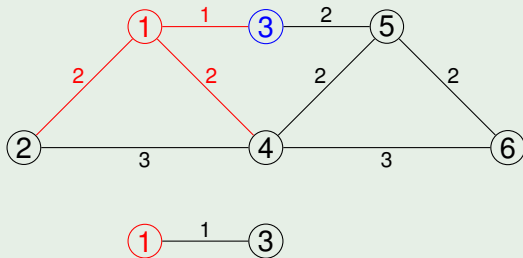
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



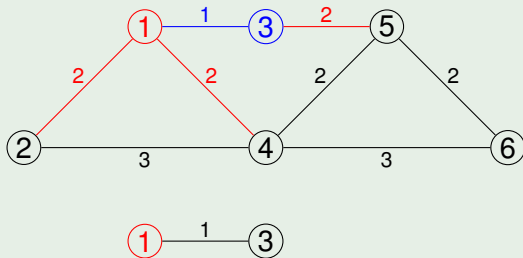
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



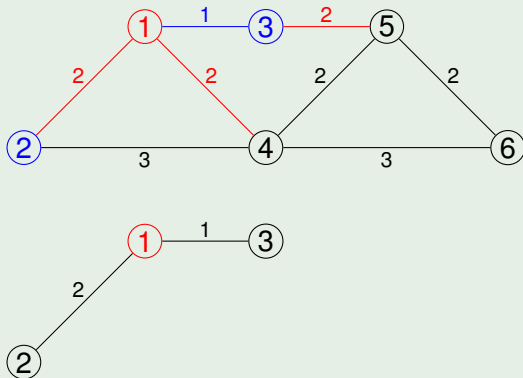
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



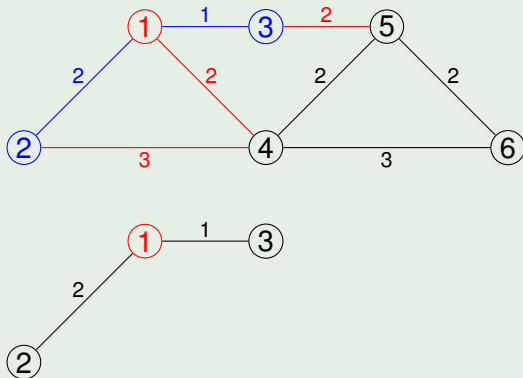
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



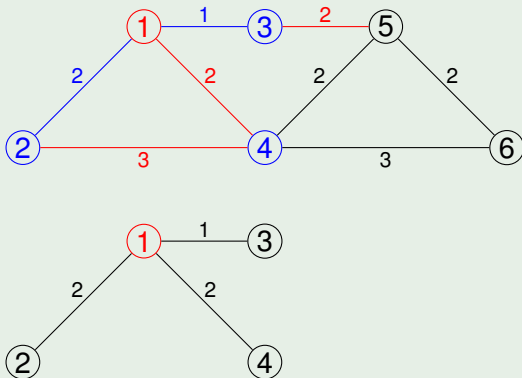
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



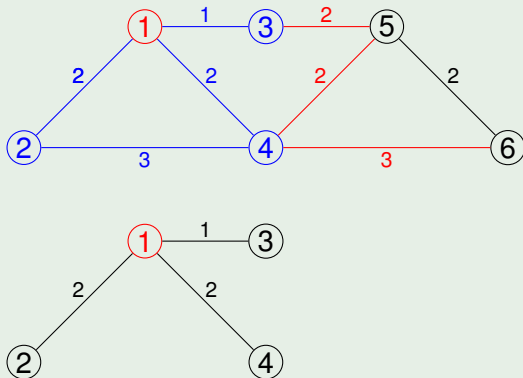
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



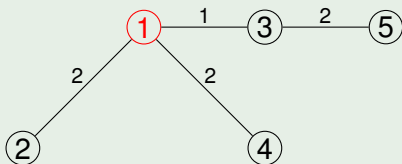
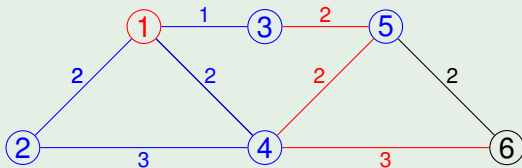
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



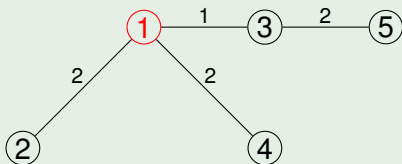
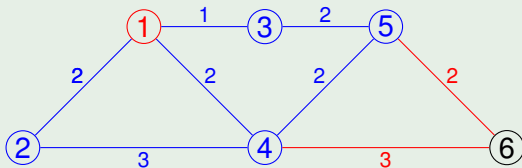
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



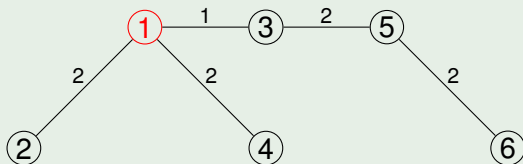
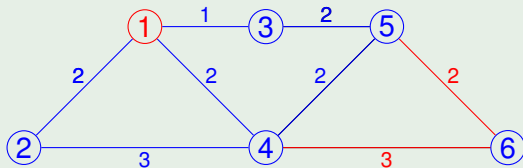
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



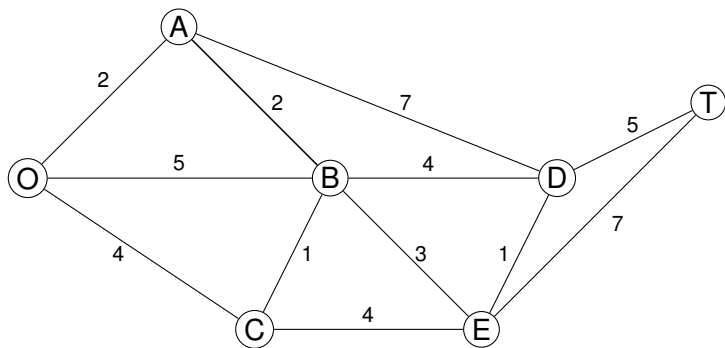
Algoritmo de Prim: ejemplo

Ejemplo (AUVM mediante Prim)



Ejercicio

Busca un AUVM utilizando el algoritmo de Kruskal y el algoritmo de Prim (usando el vértice O como punto de partida).

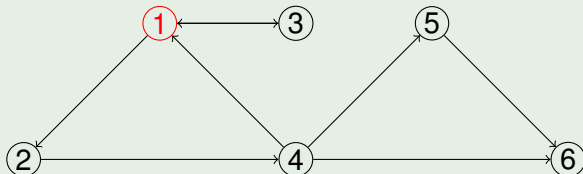


Raíz de un grafo orientado

Sea $G = (V, A)$ un grafo orientado.

- Un vértice v se dice **raíz de salida** de G si todo vértice es alcanzable desde v .

Ejemplo (1 es raíz de salida)

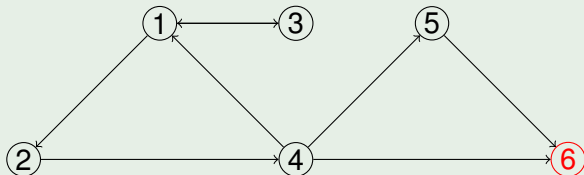


Raíz de un grafo orientado

Sea $G = (V, A)$ un grafo orientado.

- Un vértice v se dice **raíz de salida** de G si todo vértice es alcanzable desde v .
- Un vértice v se dice **raíz de entrada** de G si v es alcanzable desde todo otro vértice.

Ejemplo (6 es raíz de entrada)

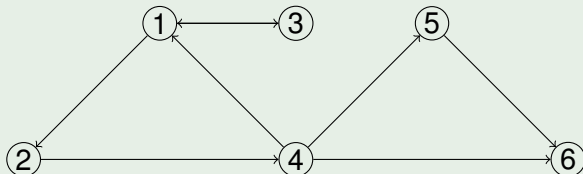


Arborescencias

Sea $G = (V, A)$ un grafo orientado.

- G es una **arborescencia de salida** si G es conexo, acíclico y tiene una raíz de salida.

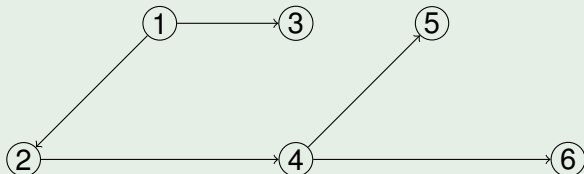
Ejemplo (G no es arborescencia)



Sea $G = (V, A)$ un grafo orientado.

- G es una **arborescencia de salida** si G es conexo, acíclico y tiene una raíz de salida.

Ejemplo (Arborescencia de salida con raíz 1)

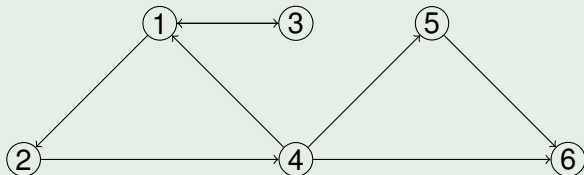


Arborescencias

Sea $G = (V, A)$ un grafo orientado.

- G es una **arborescencia de salida** si G es conexo, acíclico y tiene una raíz de salida.
- G es una **arborescencia de entrada** si G es conexo, acíclico y tiene una raíz de entrada.

Ejemplo (G no es arborescencia)

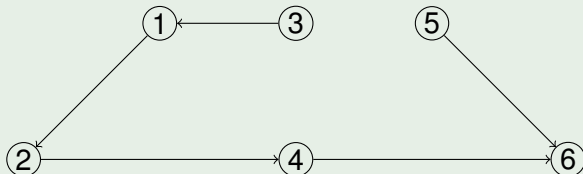


Arborescencias

Sea $G = (V, A)$ un grafo orientado.

- G es una **arborescencia de salida** si G es conexo, acíclico y tiene una raíz de salida.
- G es una **arborescencia de entrada** si G es conexo, acíclico y tiene una raíz de entrada.

Ejemplo (Arborescencia de entrada con raíz 6)

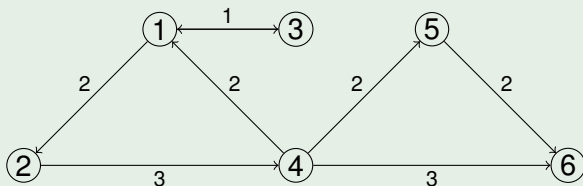


- Dado un grafo orientado $G = (V, A)$, una arborescencia de unión (de entrada/salida) es una arborescencia $G' = (V, A')$ (de entrada/salida), donde $A' \subseteq A$.
- La arborescencia de unión podría no existir.

Arborescencia de Unión

- Dado un grafo orientado $G = (V, A)$, una arborescencia de unión (de entrada/salida) es una arborescencia $G' = (V, A')$ (de entrada/salida), donde $A' \subseteq A$.
- La arborescencia de unión podría no existir.

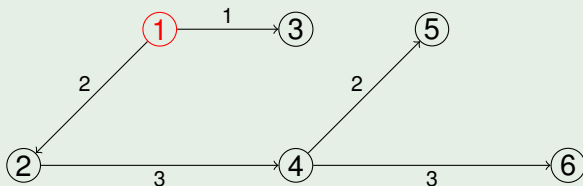
Ejemplo (Grafo orientado)



Arborescencia de Unión

- Dado un grafo orientado $G = (V, A)$, una arborescencia de unión (de entrada/salida) es una arborescencia $G' = (V, A')$ (de entrada/salida), donde $A' \subseteq A$.
- La arborescencia de unión podría no existir.

Ejemplo (Arborescencia de unión con raíz 1)



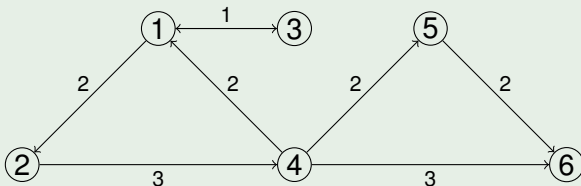
Arborescencia de Unión de Valor Mínimo

- Partimos de una red orientada $R = (V, A, p)$.
- Buscamos una red orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es una arborescencia de unión del grafo $G = (V, A)$, y de manera que minimiza $\sum_{e \in A'} p_e$.

Arborescencia de Unión de Valor Mínimo

- Partimos de una red orientada $R = (V, A, p)$.
- Buscamos una red orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es una arborescencia de unión del grafo $G = (V, A)$, y de manera que minimiza $\sum_{e \in A'} p_e$.

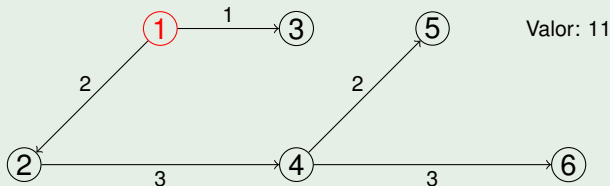
Ejemplo (Grafo orientado)



Arborescencia de Unión de Valor Mínimo

- Partimos de una red orientada $R = (V, A, p)$.
- Buscamos una red orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es una arborescencia de unión del grafo $G = (V, A)$, y de manera que minimiza $\sum_{e \in A'} p_e$.

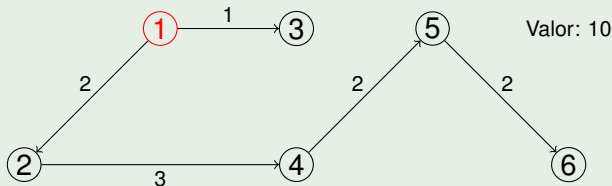
Ejemplo (Arborescencia de unión con raíz 1, ejemplo 1)



Arborescencia de Unión de Valor Mínimo

- Partimos de una red orientada $R = (V, A, p)$.
- Buscamos una red orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es una arborescencia de unión del grafo $G = (V, A)$, y de manera que minimiza $\sum_{e \in A'} p_e$.

Ejemplo (Arborescencia de unión con raíz 1, ejemplo 2)



- **AUVM fijado una raíz de salida:** Buscamos una red orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es una arborescencia de unión del grafo $G = (V, A)$ con raíz de salida v de manera que minimiza $\sum_{e \in A'} p_e$.
- **AUVM fijado una raíz de entrada:** Buscamos una red orientada $R' = (V, A', p)$ de manera que $G' = (V, A')$ es una arborescencia de unión del grafo $G = (V, A)$ con raíz de entrada v de manera que minimiza $\sum_{e \in A'} p_e$.