

# Video 44 (1)

---

## Conceptos Principales

---

1. a
2. b
3. c

## Notas

---

### Árbol de Expansión Mínima (Kruskal)

---

#### Resumen para estudiar

El algoritmo de **Kruskal** es otro método para obtener el **árbol de expansión mínima (MST)** de un grafo no dirigido y valorado.

A diferencia de Prim, Kruskal **no comienza en un vértice**, sino que trabaja **directamente con las aristas**, ordenándolas por peso y agregándolas mientras no formen ciclos.

Es muy usado en redes, telecomunicaciones y problemas de infraestructura, porque encuentra la forma más barata de conectar todos los nodos.

---

#### Idea principal

1. Tomar **todas las aristas** del grafo.
2. Ordenarlas por **peso ascendente**.
3. Ir agregándolas una por una al MST **si no forman un ciclo**.
4. Parar cuando todos los vértices están en **una sola componente conexa** (o cuando se hayan agregado  $n-1$  aristas).

Kruskal siempre construye el MST desde las aristas más baratas hacia arriba, verificando componentes conexas para evitar ciclos.

---

### Cómo funciona Kruskal paso a paso

---

#### 1. Listar todas las aristas

Se arma una tabla con cada arista y su peso, por ejemplo:

- A–J peso 1
- J–B peso 2
- I–U peso 2

- etc.

Este formato es más cómodo porque permite ordenar fácilmente.

## 2. Ordenar aristas de menor a mayor

Kruskal siempre evalúa primero las más baratas.

Esta es la esencia del algoritmo.

## 3. Crear componentes conexas

Cada arista que se acepta se coloca dentro de una **componente conexa**.

Conceptualmente, la componente conexa es un conjunto o estructura que agrupa vértices conectados.

En implementación real, pueden ser:

- Listas de adyacencia
- Arreglos
- Estructura **Union-Find** (lo óptimo)

En el video, el profesor lo hace visualmente:

cuando toma una arista como J-B, crea una pequeña lista donde pone J y B como conectados.

## 4. Recorrer aristas en orden ascendente

Para cada arista:

- Si sus dos vértices están en **componentes distintas**, se une y se combinan esas dos componentes.
- Si están en la **misma componente**, esa arista **se descarta** porque crearía un ciclo.

Ejemplo del video:

- Las aristas con peso 2 (J-B y I-U) se agregan sin problema.
- Cuando aparece una arista donde ambos vértices ya están en la misma componente, se omite.

Esto evita ciclos automáticamente.

## 5. Seguir avanzando hasta que todos los vértices estén conectados

El algoritmo termina cuando:

- Todos los vértices están en **una sola componente**, o
- Se han agregado **n-1 aristas**.

Si al final quedan aristas sin usar, es normal: simplemente son aristas que causarían ciclos.

## 6. Obtener el MST final

Se listan las aristas que sí se usaron y se calcula su peso total.

En el video:

- El MST obtenido por Kruskal es equivalente al que se obtuvo por Prim, confirmando que ambos algoritmos llegan al mismo costo mínimo.
- El valor total era aproximadamente **80** (o 76 según el ajuste de un peso que se corrigió).