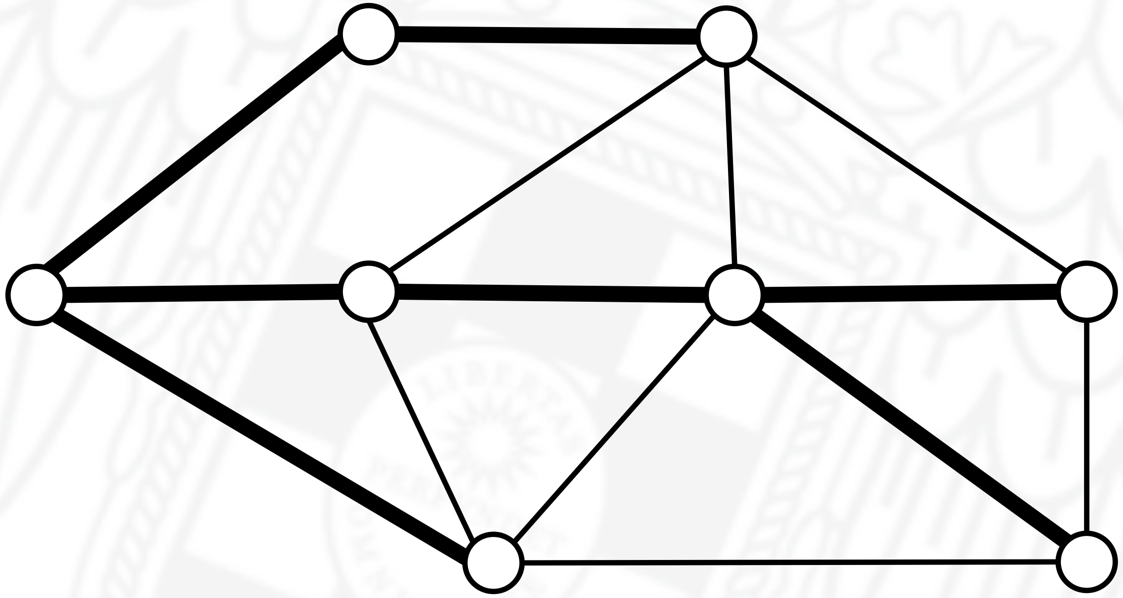
ÁRBOLES DE RECUBRIMIENTO DE COSTE MÍNIMO

UNIVERSIDAD COMPLUTENSE MADRID

ALBERTO VERDEJO

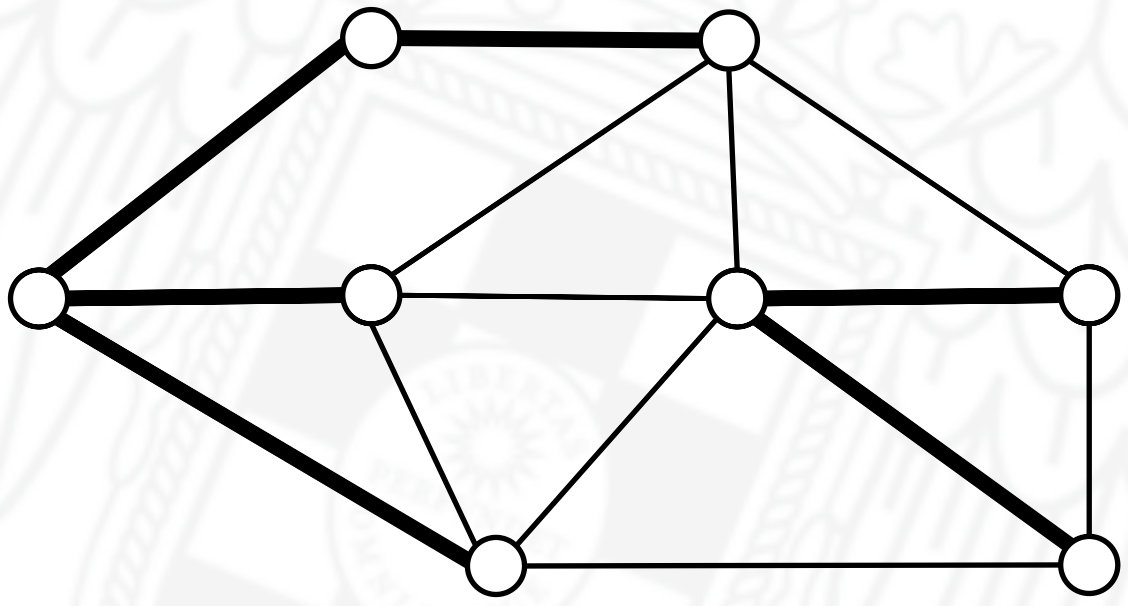
Árbol de recubrimiento

- Dado un grafo no dirigido G, un árbol de recubrimiento de G es un subgrafo T tal que:
  - T es un árbol: es conexo y acíclico
  - T es de recubrimiento: alcanza todos los vértices de G



Árbol de recubrimiento

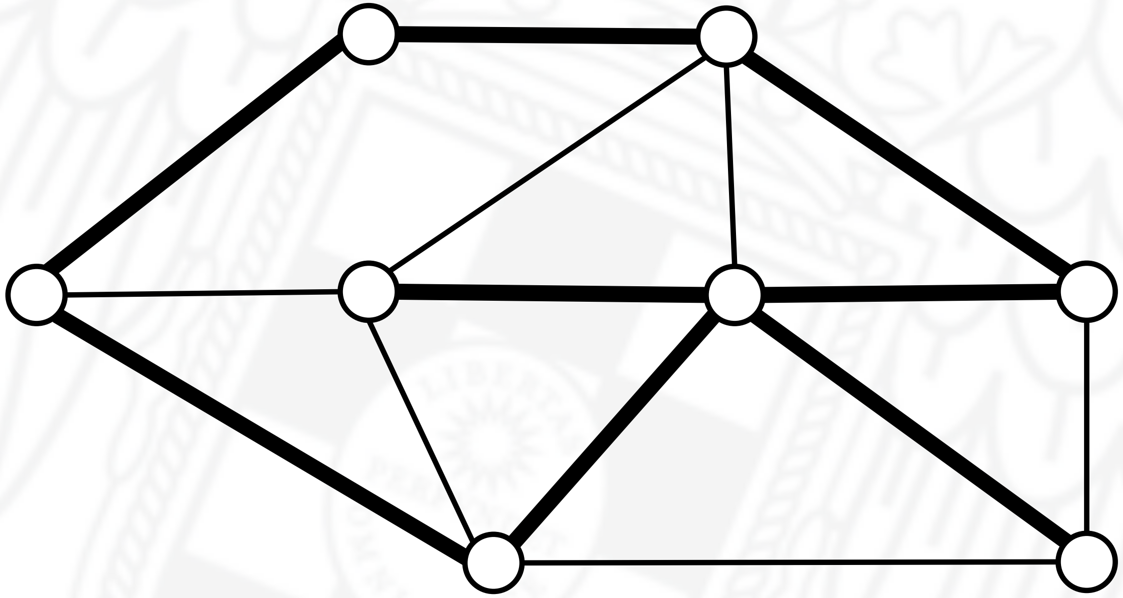
- Dado un grafo no dirigido G, un árbol de recubrimiento de G es un subgrafo T tal que:
  - T es un árbol: es conexo y acíclico
  - T es de recubrimiento: alcanza todos los vértices de G



no es conexo

Árbol de recubrimiento

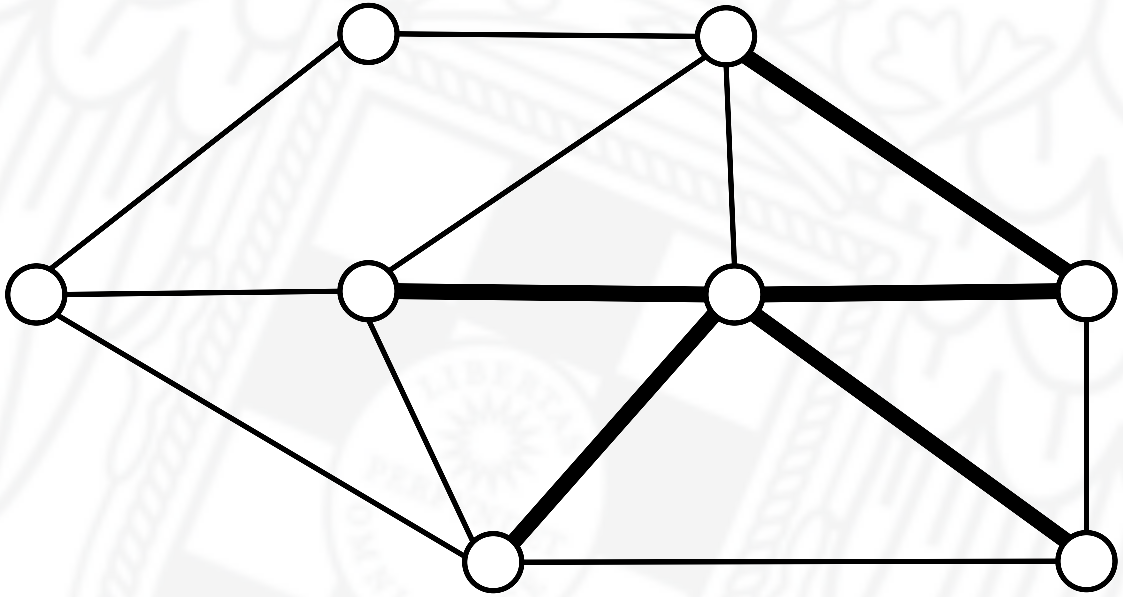
- Dado un grafo no dirigido G, un árbol de recubrimiento de G es un subgrafo T tal que:
  - T es un árbol: es conexo y acíclico
  - T es de recubrimiento: alcanza todos los vértices de G



no es un árbol (tiene ciclos)

Árbol de recubrimiento

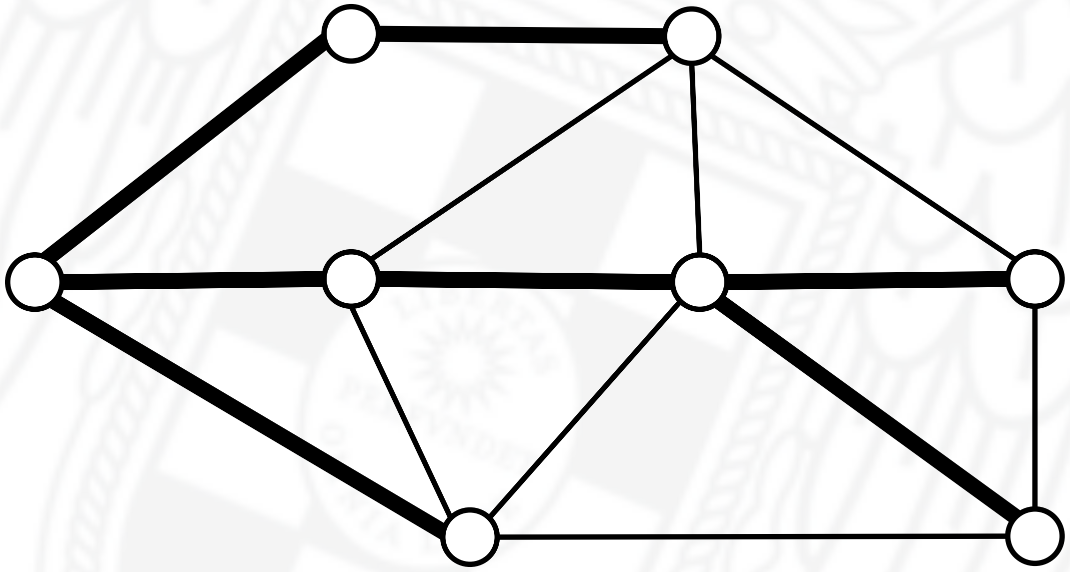
- Dado un grafo no dirigido *G*, un árbol de recubrimiento de *G* es un subgrafo *T* tal que:
  - *T* es un árbol: es conexo y acíclico
  - *T* es de recubrimiento: alcanza todos los vértices de G



no es de recubrimiento

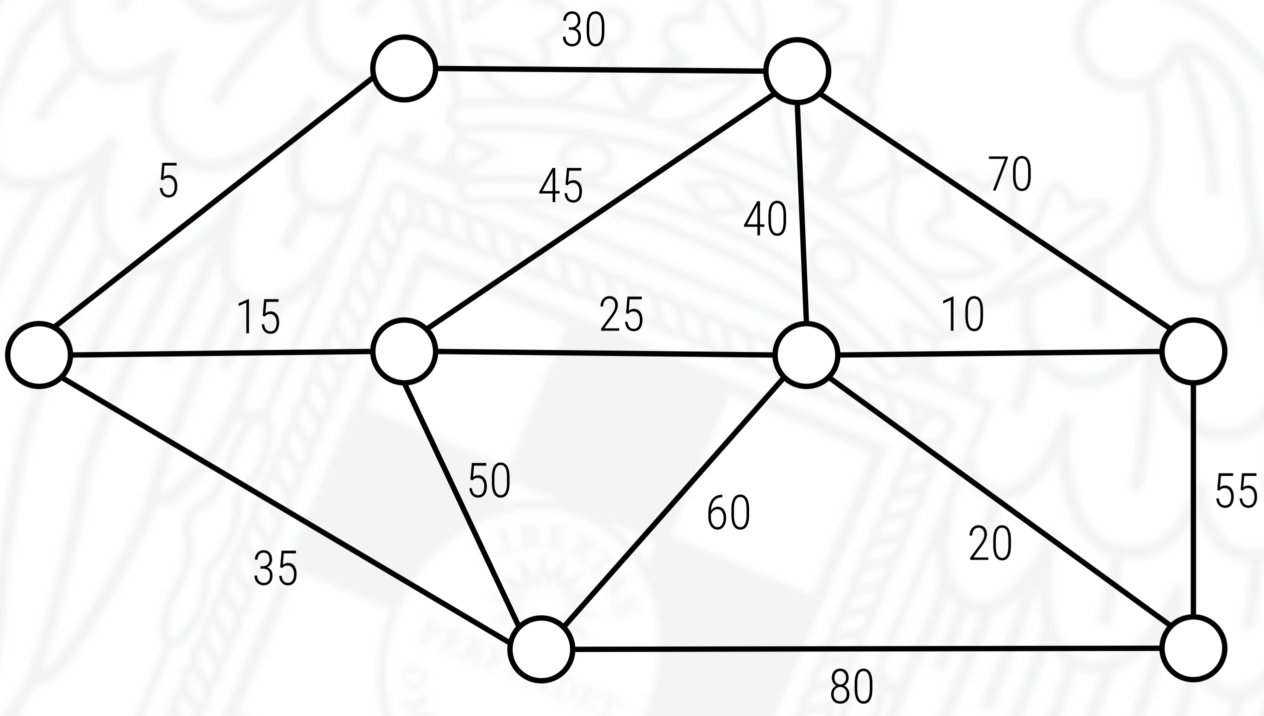
Árbol de recubrimiento

- Si T es un árbol de recubrimiento de un grafo G con V vértices:
  - T contiene exactamente $V - 1$ aristas.
  - Al eliminar cualquier arista de T deja de ser conexo.
  - Añadir cualquier arista a T crea un ciclo.



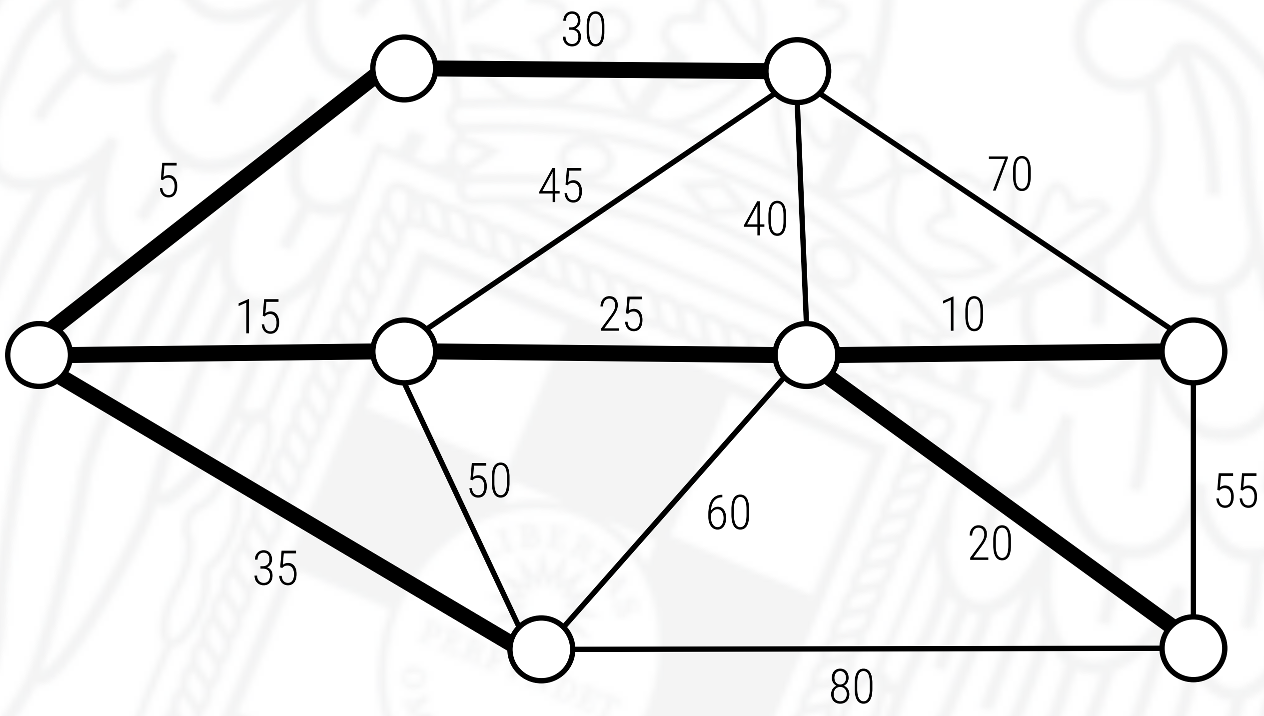
Problema del árbol de recubrimiento de coste mínimo

- Dado un grafo valorado no dirigido G, encontrar un árbol de recubrimiento de coste mínimo (ARM).



Problema del árbol de recubrimiento de coste mínimo

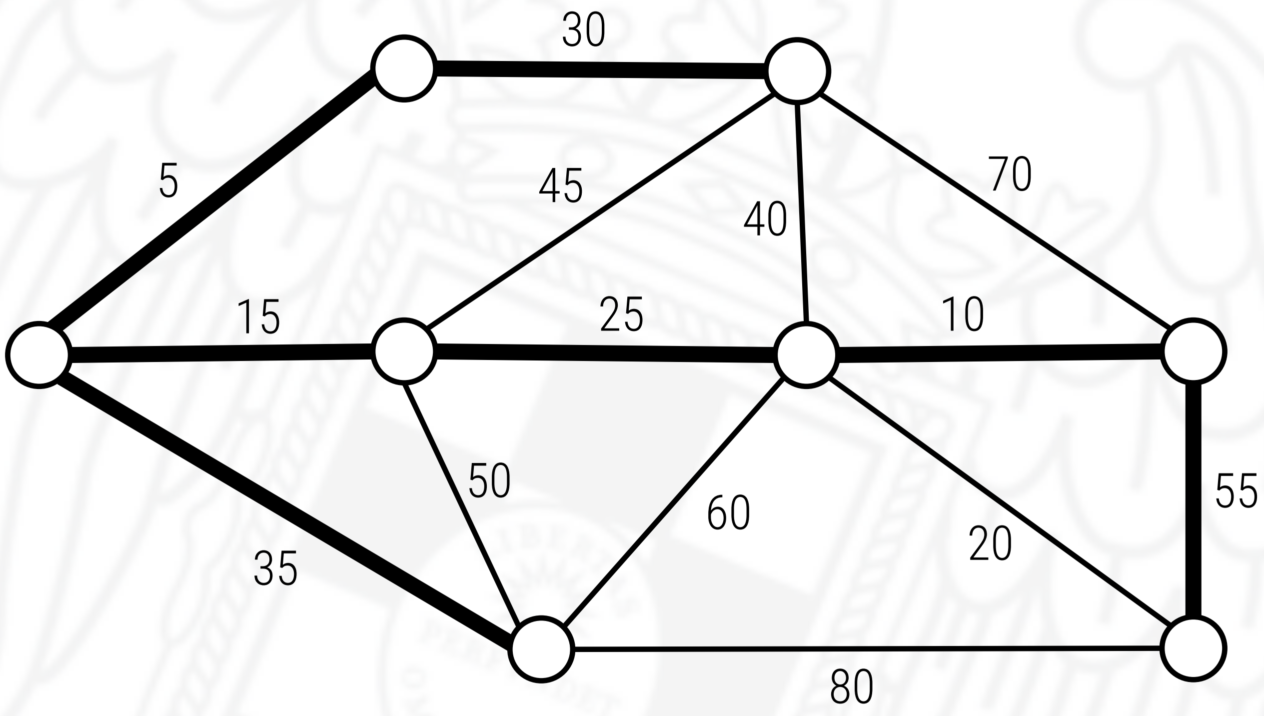
- Dado un grafo valorado no dirigido G, encontrar un árbol de recubrimiento de coste mínimo (ARM).



5 + 15 + 35 + 30 + 25 + 10 + 20 = 140

Problema del árbol de recubrimiento de coste mínimo

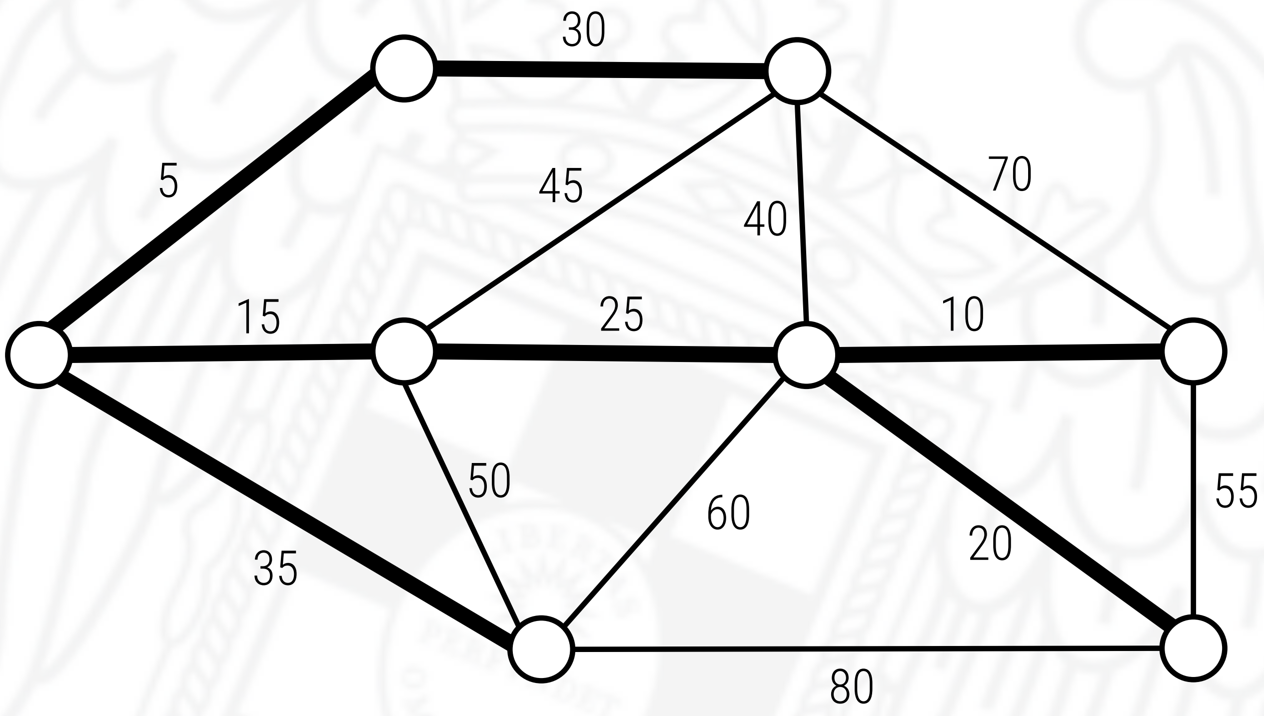
- Dado un grafo valorado no dirigido *G*, encontrar un árbol de recubrimiento de coste mínimo (ARM).



no es mínimo (coste 175)

Problema del árbol de recubrimiento de coste mínimo

- Dado un grafo valorado no dirigido G, encontrar un árbol de recubrimiento de coste mínimo (ARM).

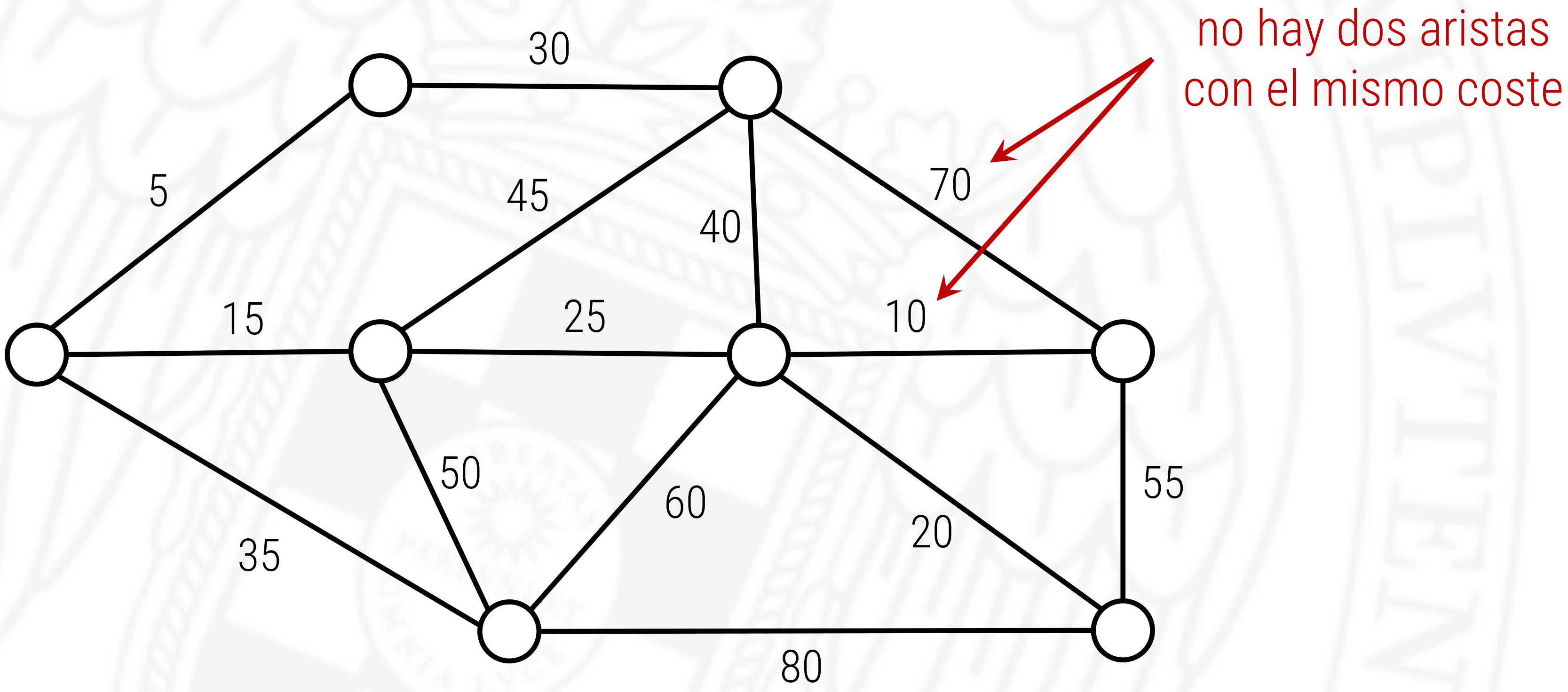


Aplicaciones de los ARMs

- Verificación facial en tiempo real.
- Búsqueda de redes de carreteras en imágenes de satélites o aéreas.
- Reducción del almacenamiento de datos en la secuenciación de aminoácidos de una proteína.
- Modelar la localidad de interacciones entre partículas en flujos de fluidos turbulentos.
- Algoritmos de aproximación para problemas NP-difíciles (por ejemplo, TSP, árbol de Steiner).
- Diseño de redes (comunicaciones, eléctricas, hidráulicas, informáticas, viales).

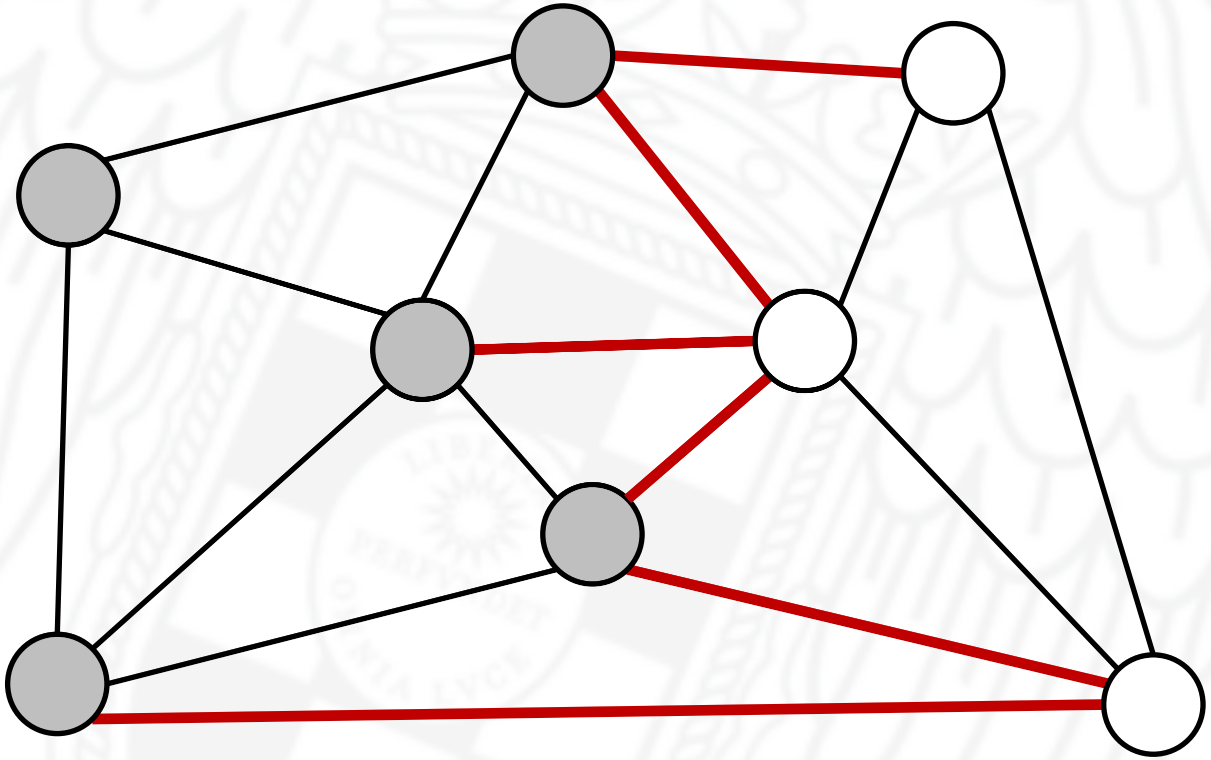
Dos simplificaciones

- El grafo es conexo $\implies$ existe un ARM
- Los valores de las aristas son todos distintos $\implies$ ARM único



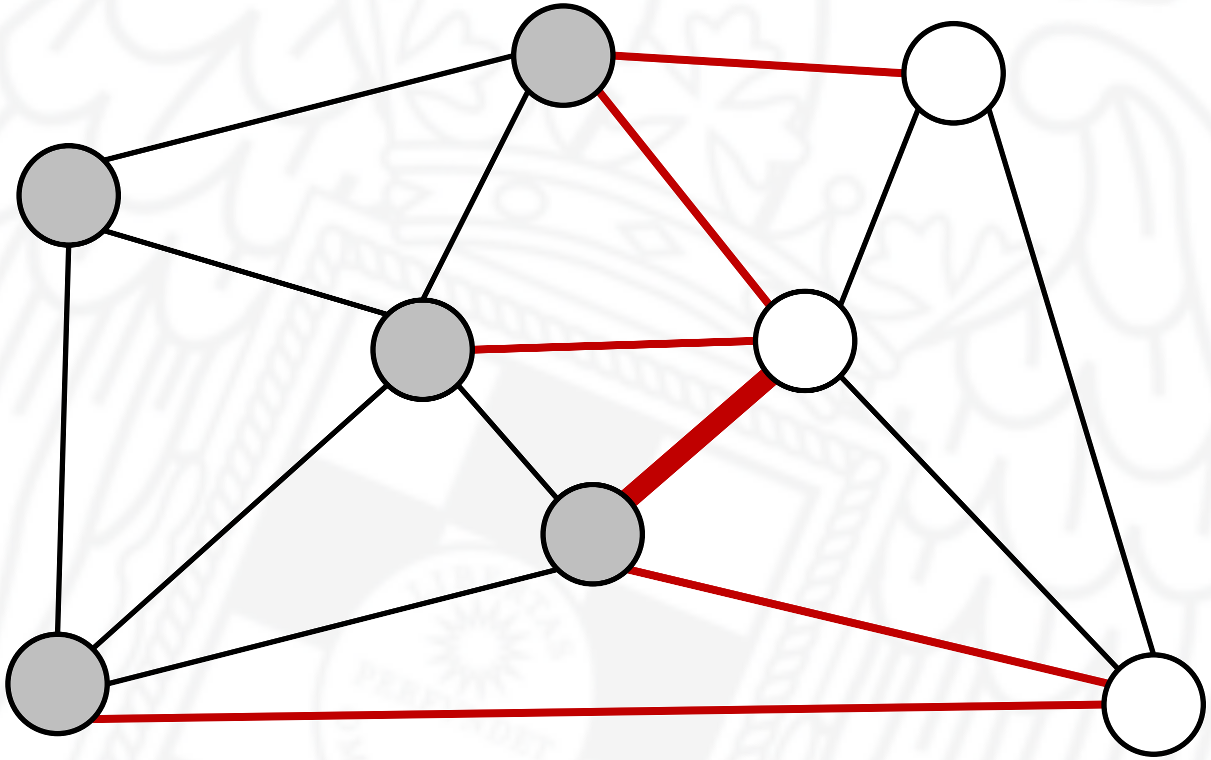
La propiedad del corte

- Un corte de un grafo es una partición de sus vértices en dos conjuntos no vacíos.
- Una arista cruza el corte si tiene un extremo en cada conjunto.



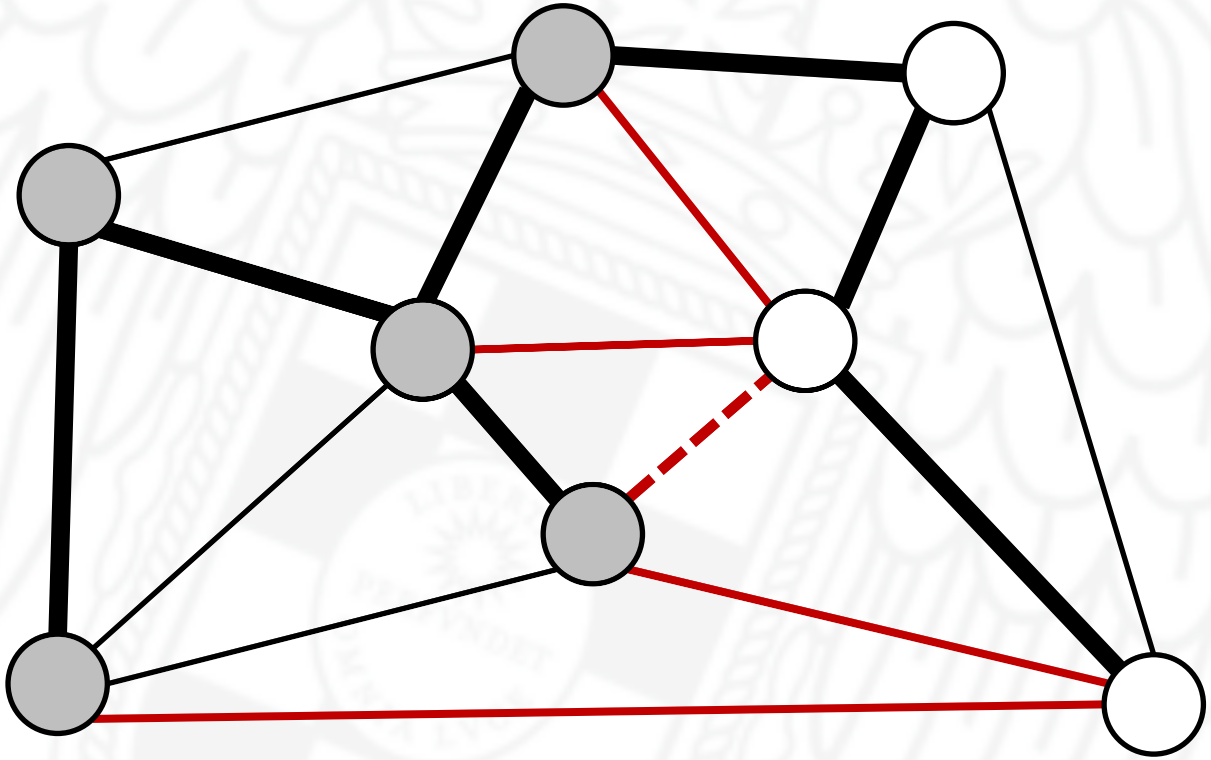
La propiedad del corte

- La propiedad del corte: dado un corte cualquiera, la arista de menor peso que lo cruza pertenece al ARM.



La propiedad del corte

- La propiedad del corte: dado un corte cualquiera, la arista de menor peso que lo cruza pertenece al ARM.
- Demostración:

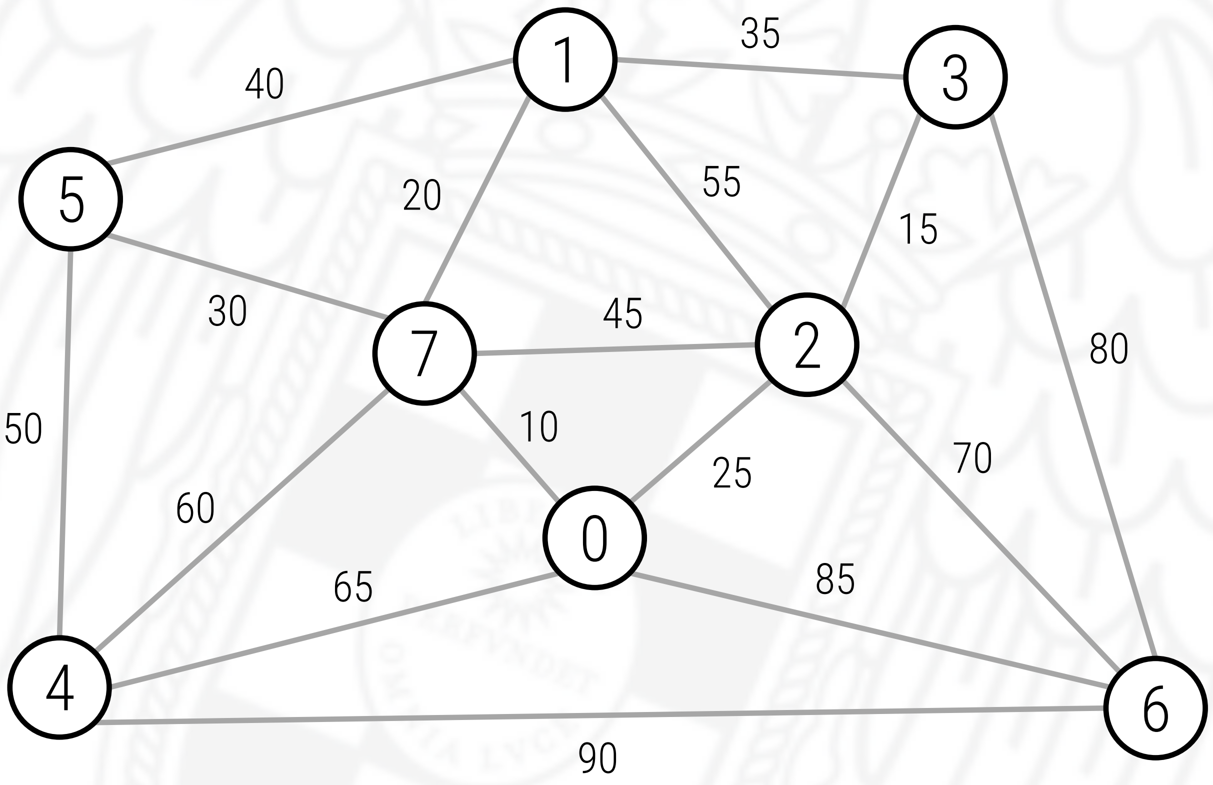


Árbol de recubrimiento de coste mínimo

```
template <typename Valor>
class ARM {

public:

   ARM(GrafoValorado<Valor> const& g);

   Valor costeARM() const;

   std::vector<Arista<Valor>> ARM() const;

};
```

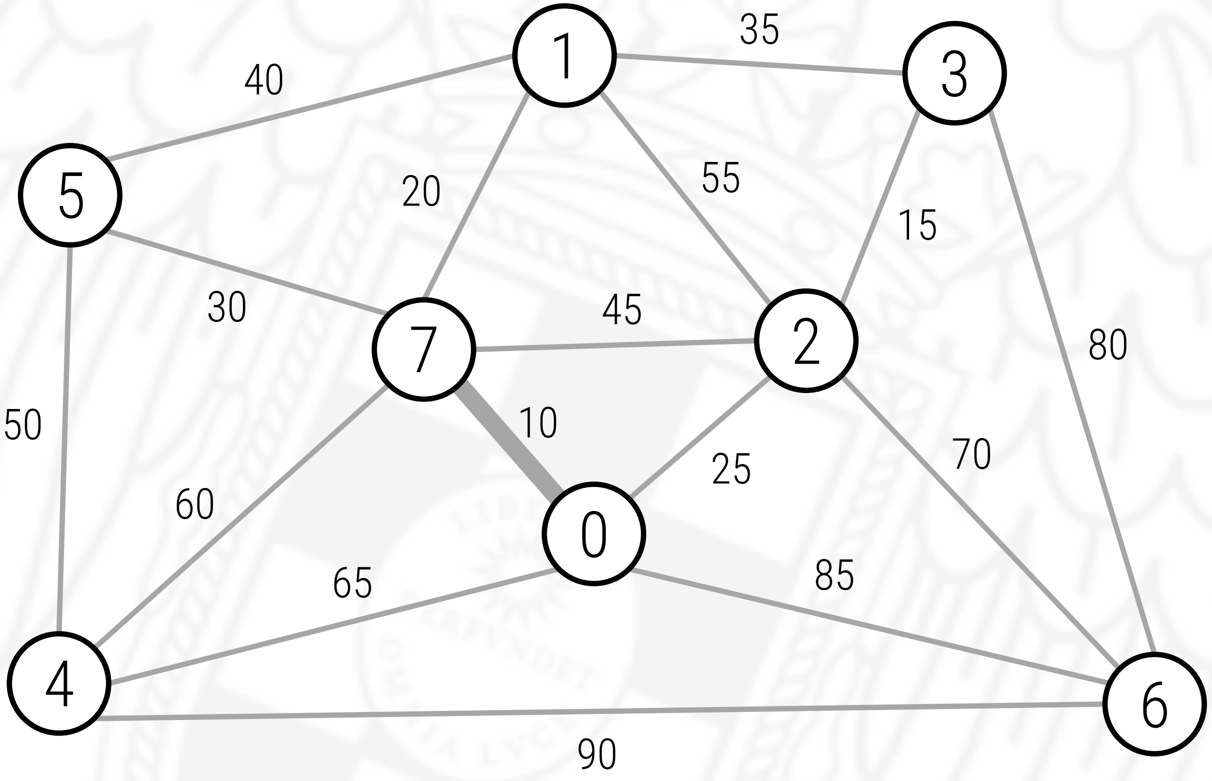
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



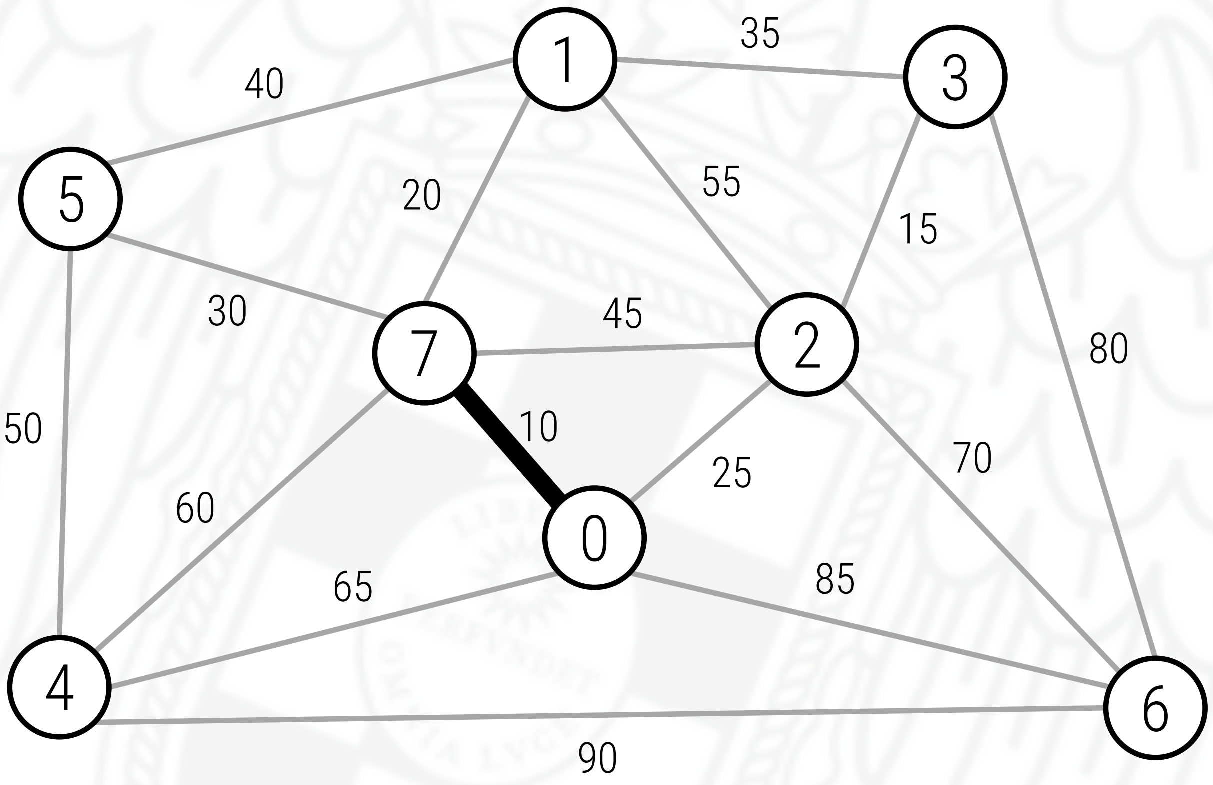
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



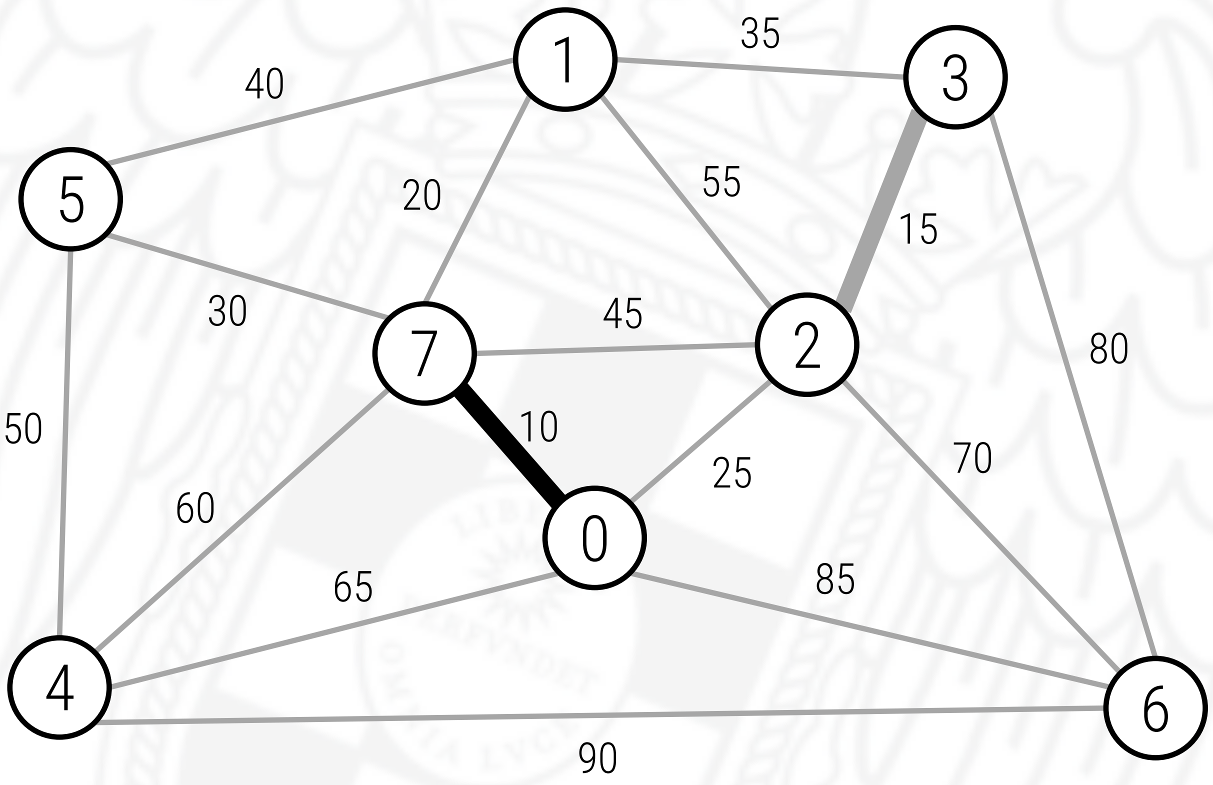
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



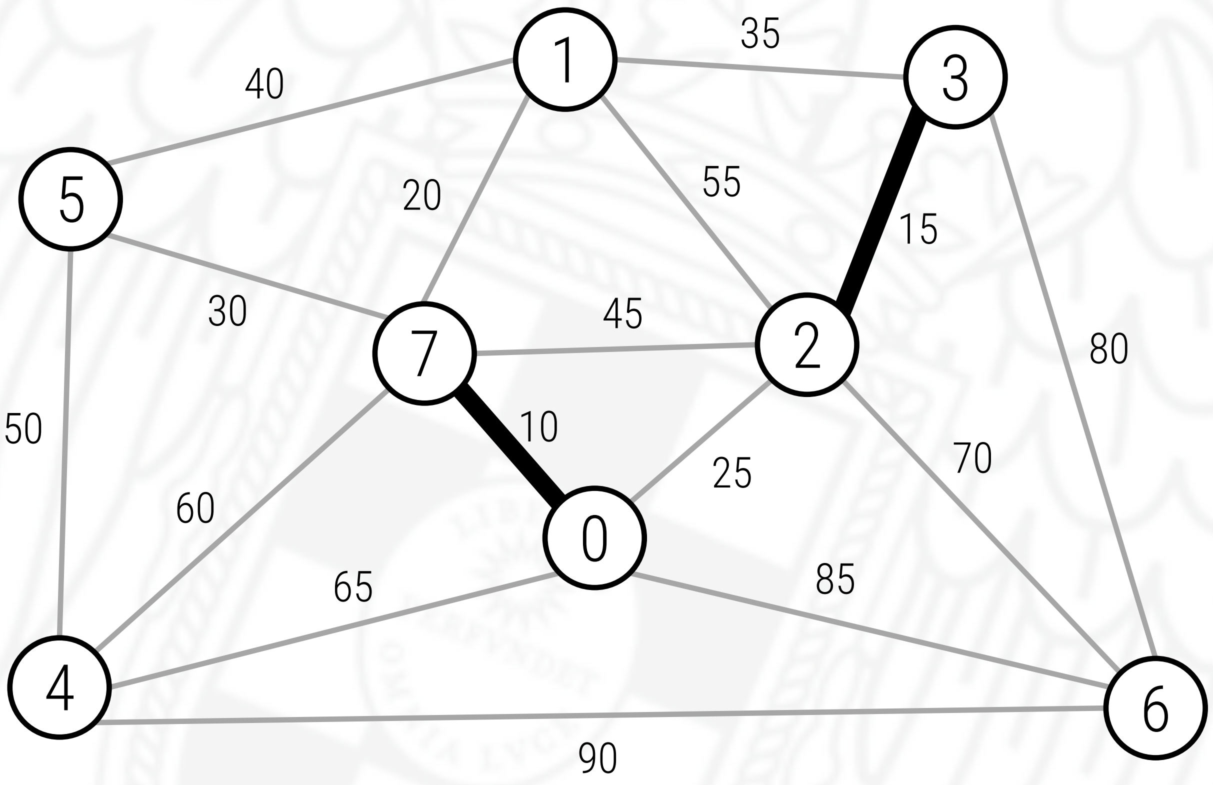
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



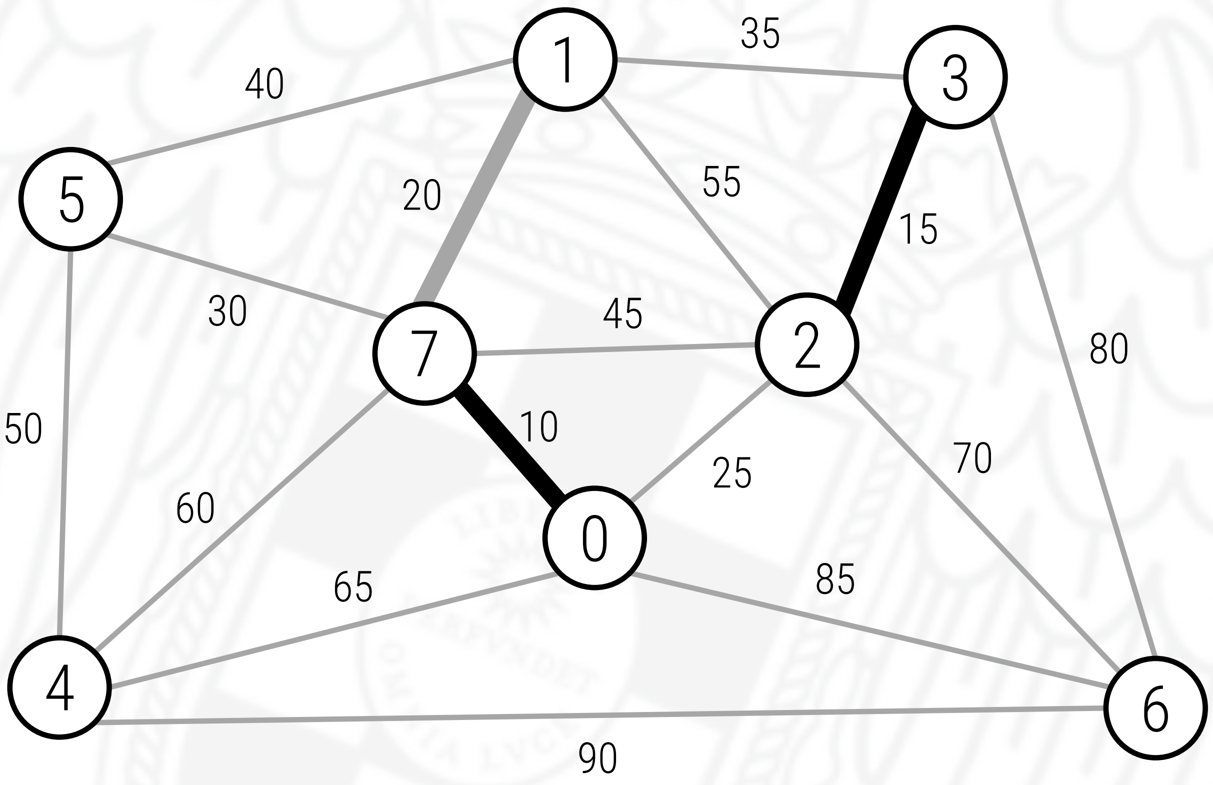
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



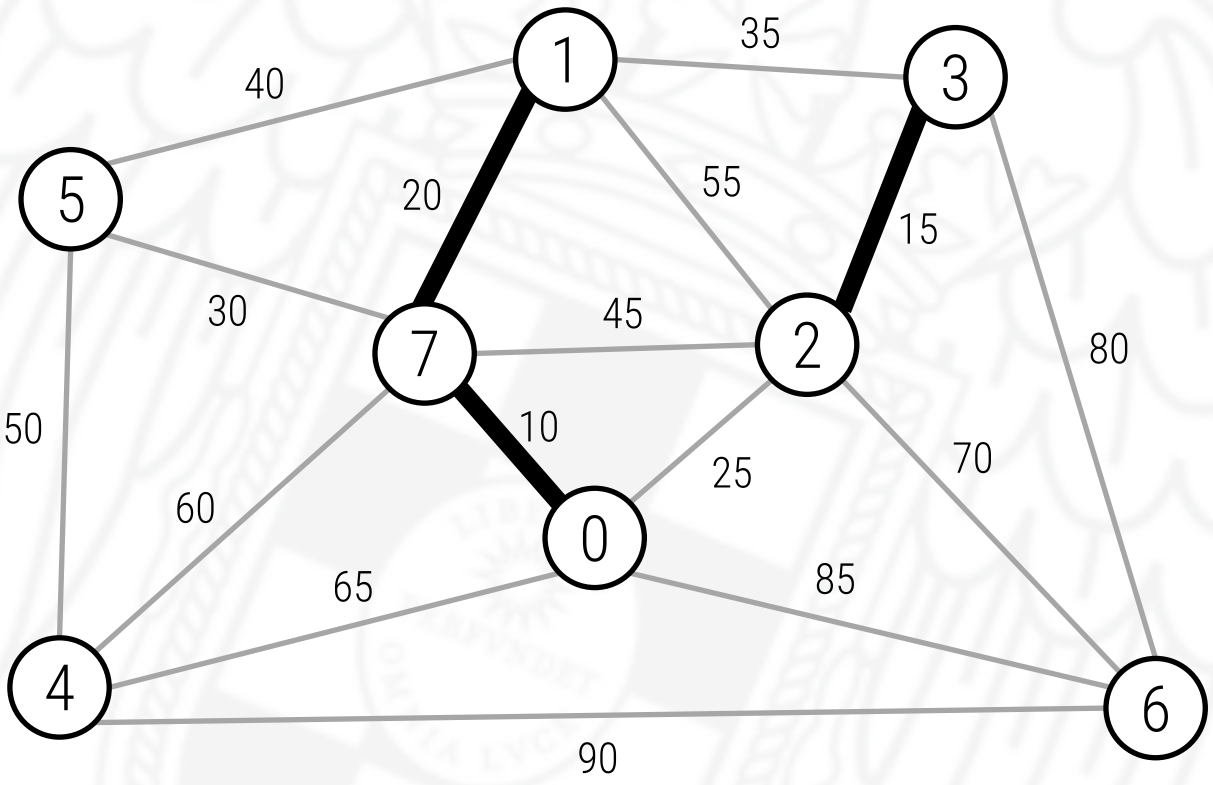
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



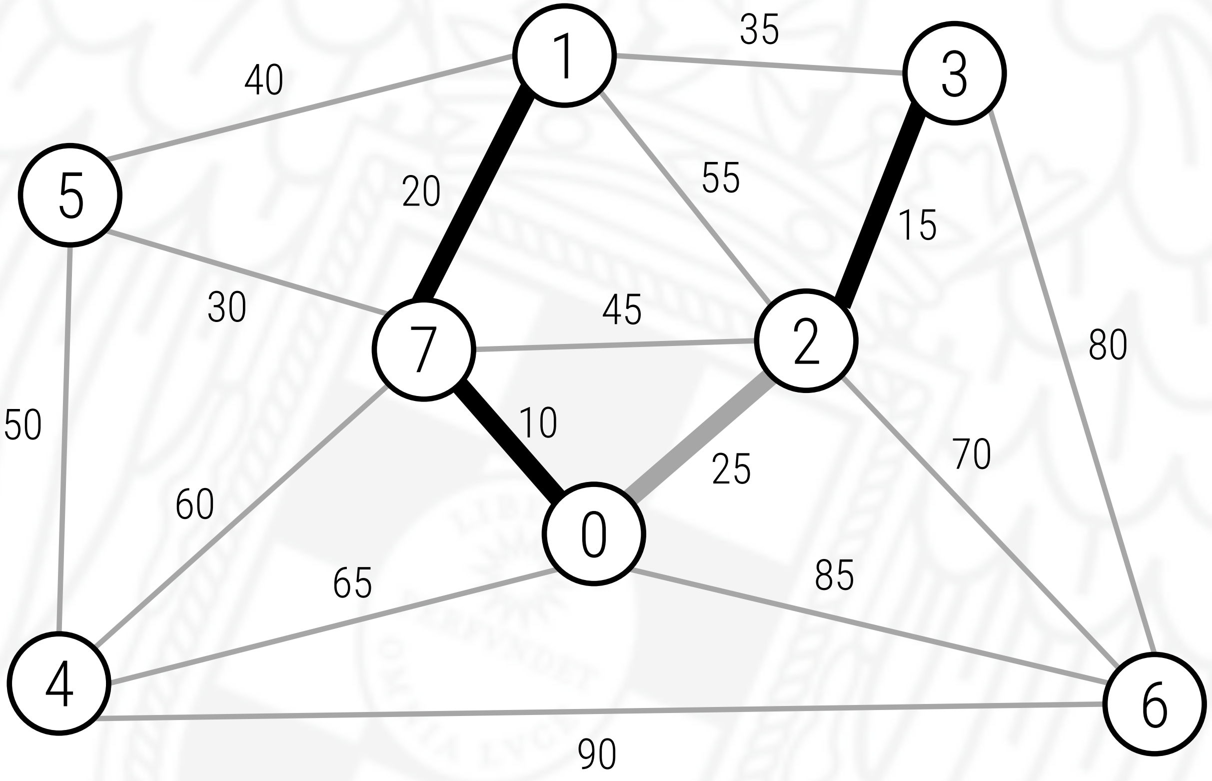
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



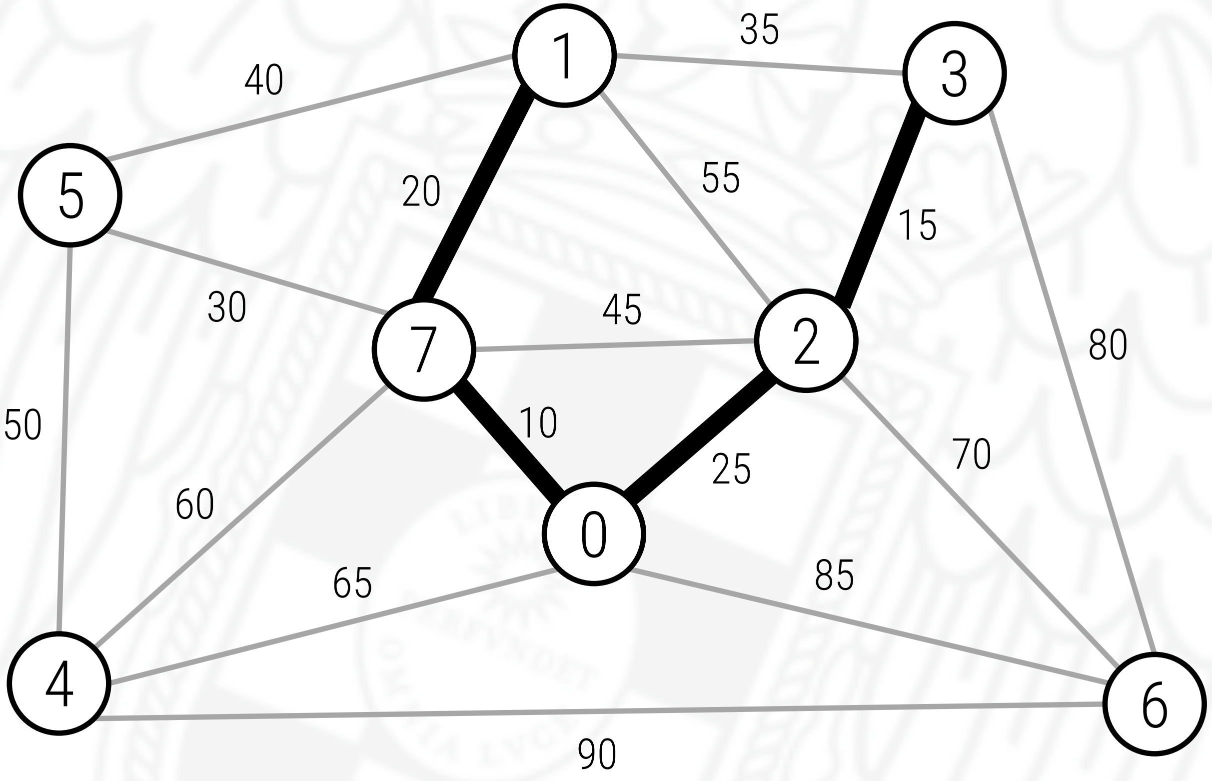
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



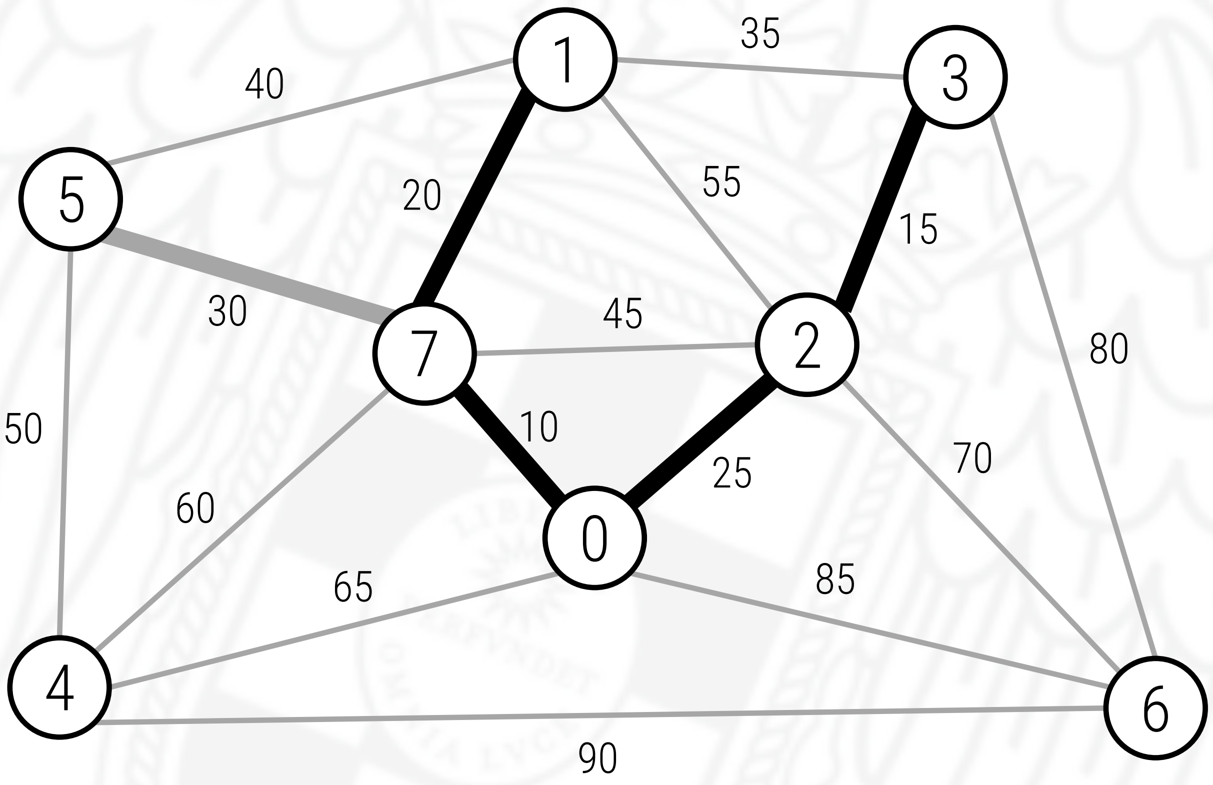
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



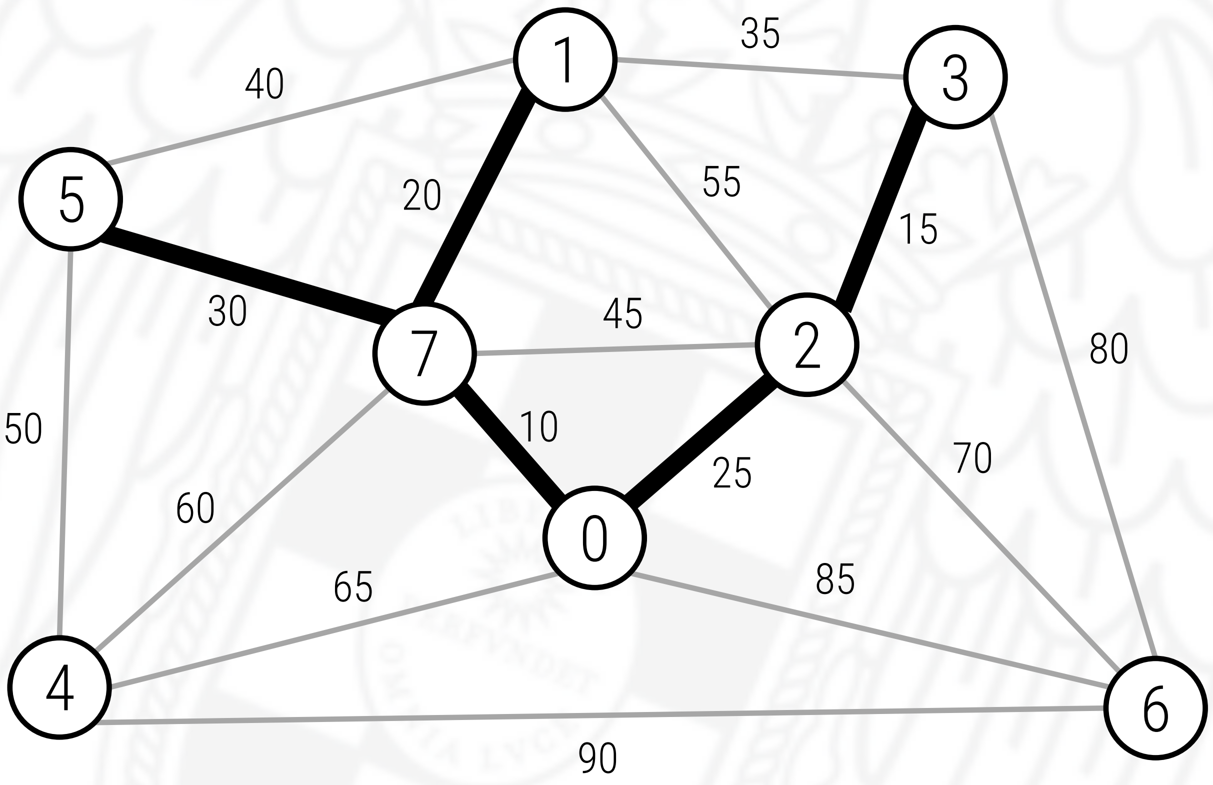
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



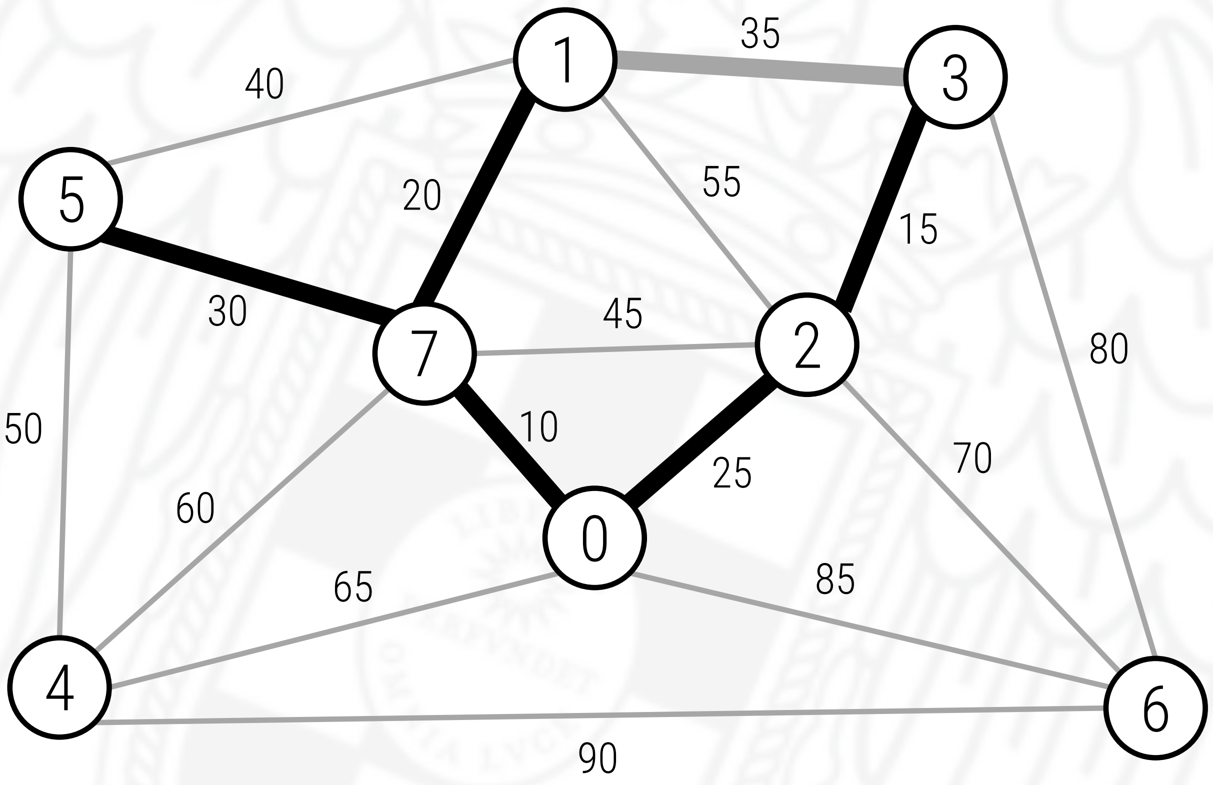
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



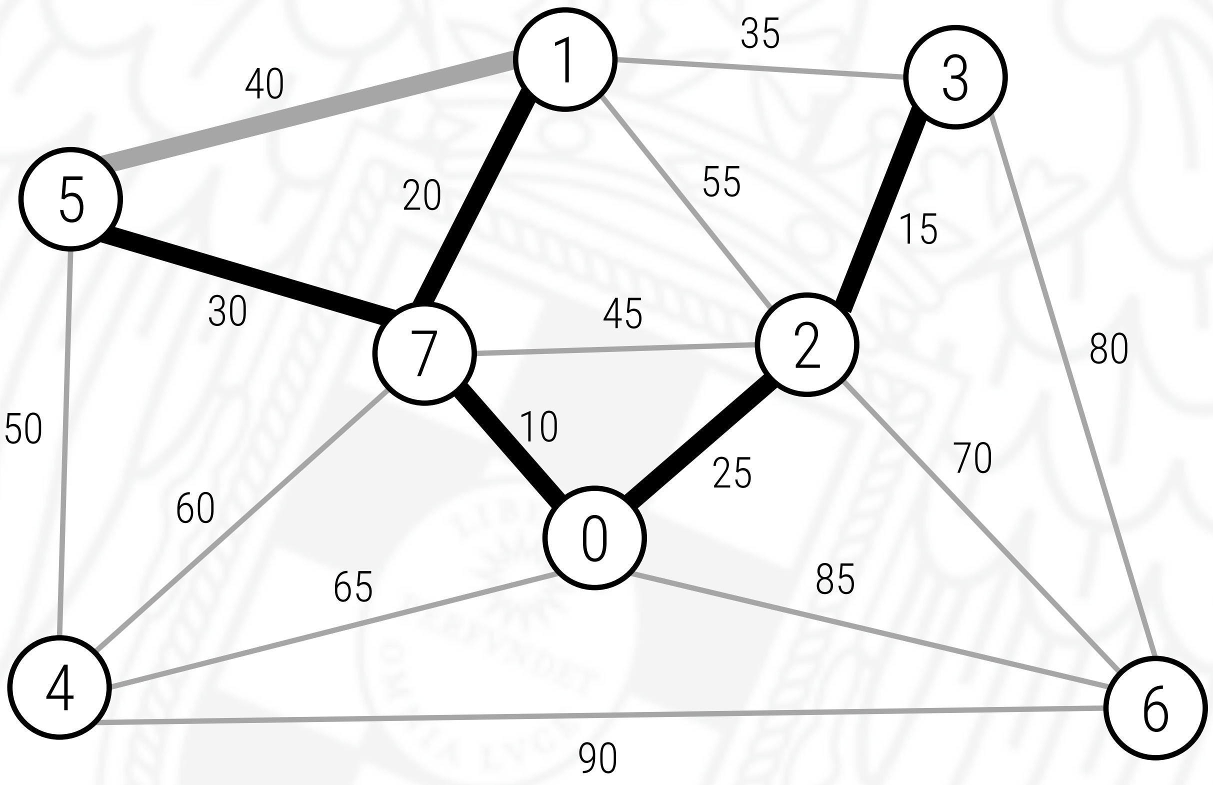
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



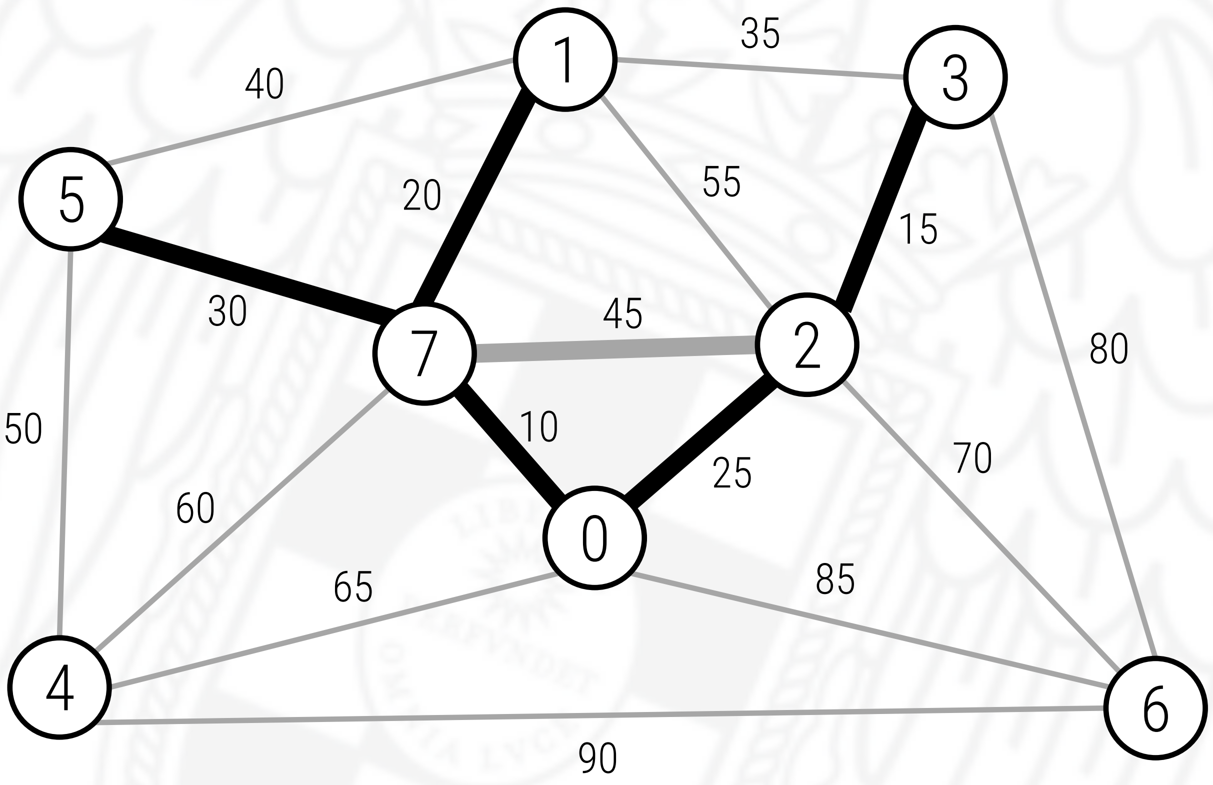
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



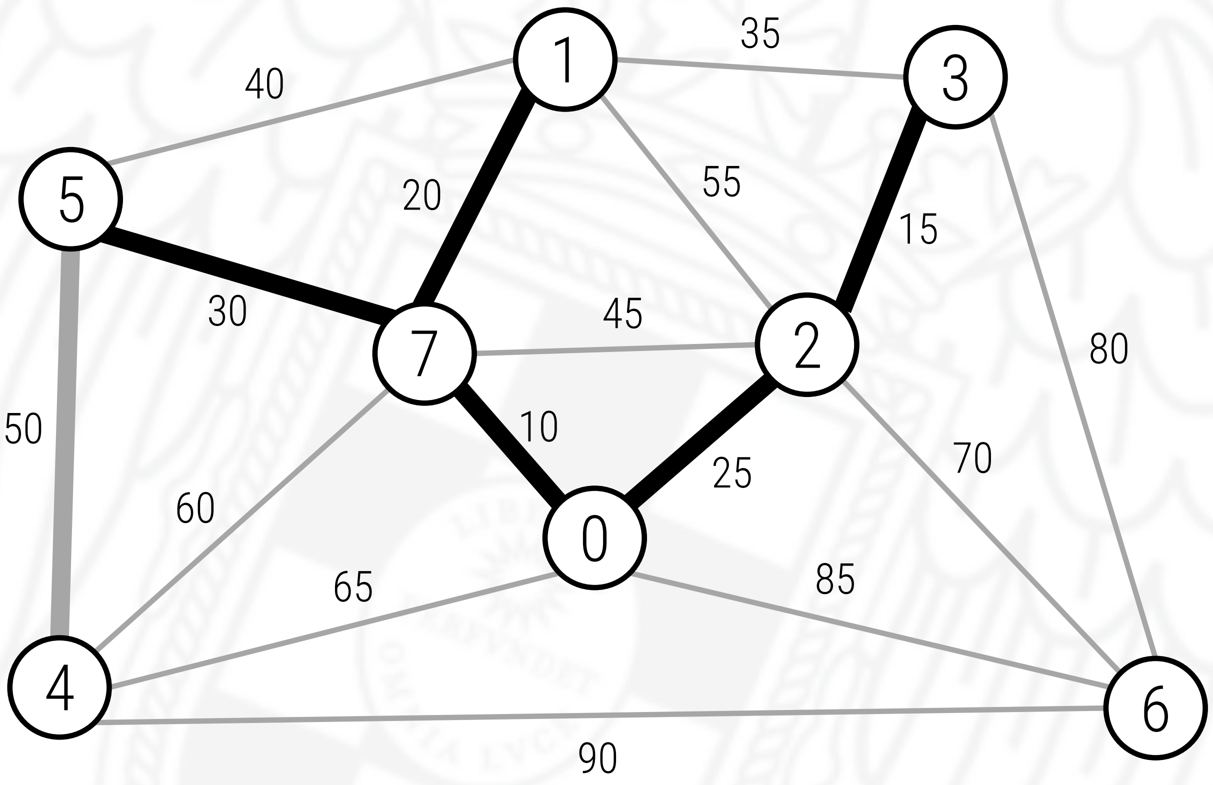
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



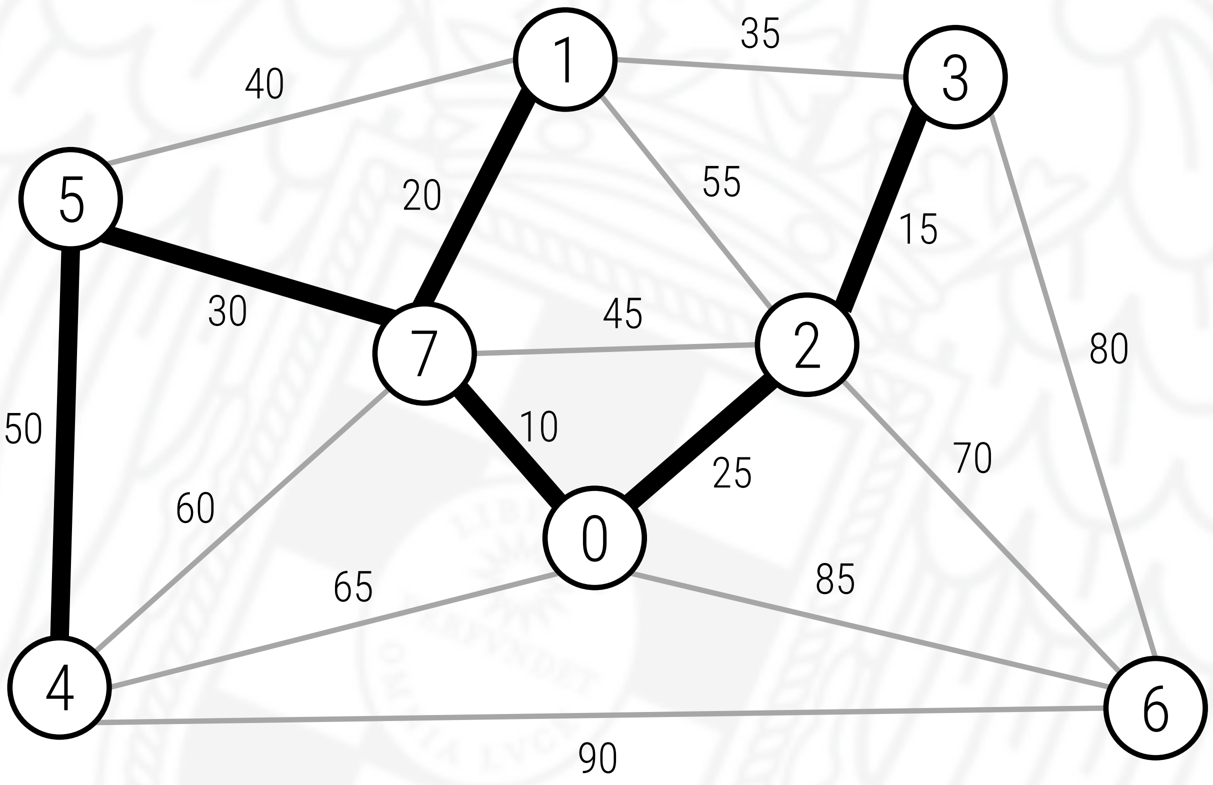
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



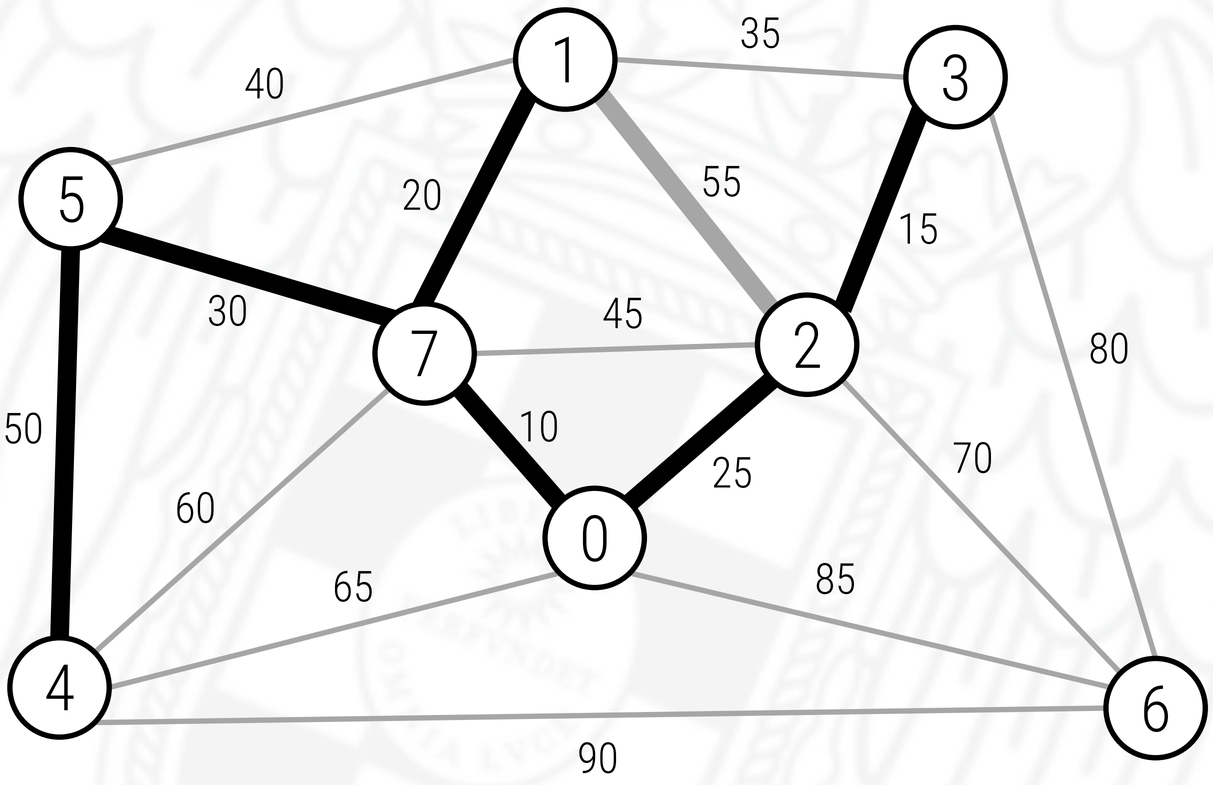
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



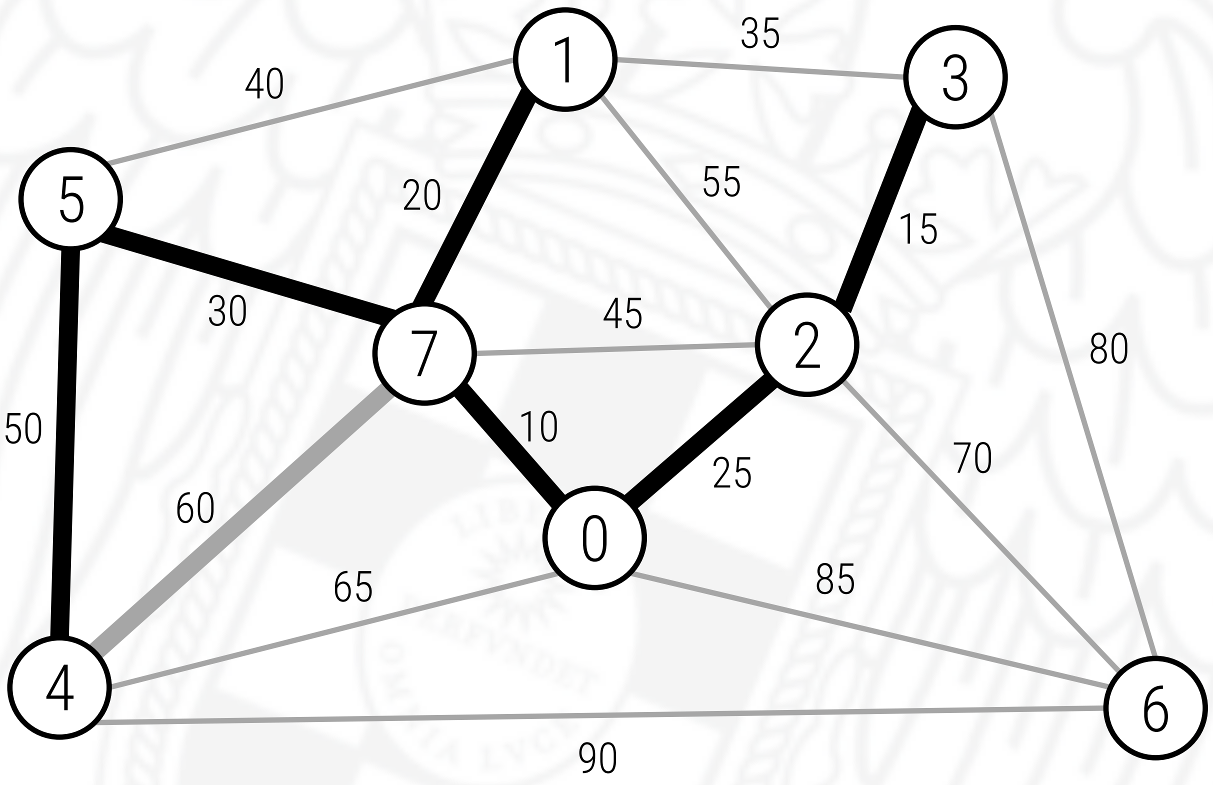
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



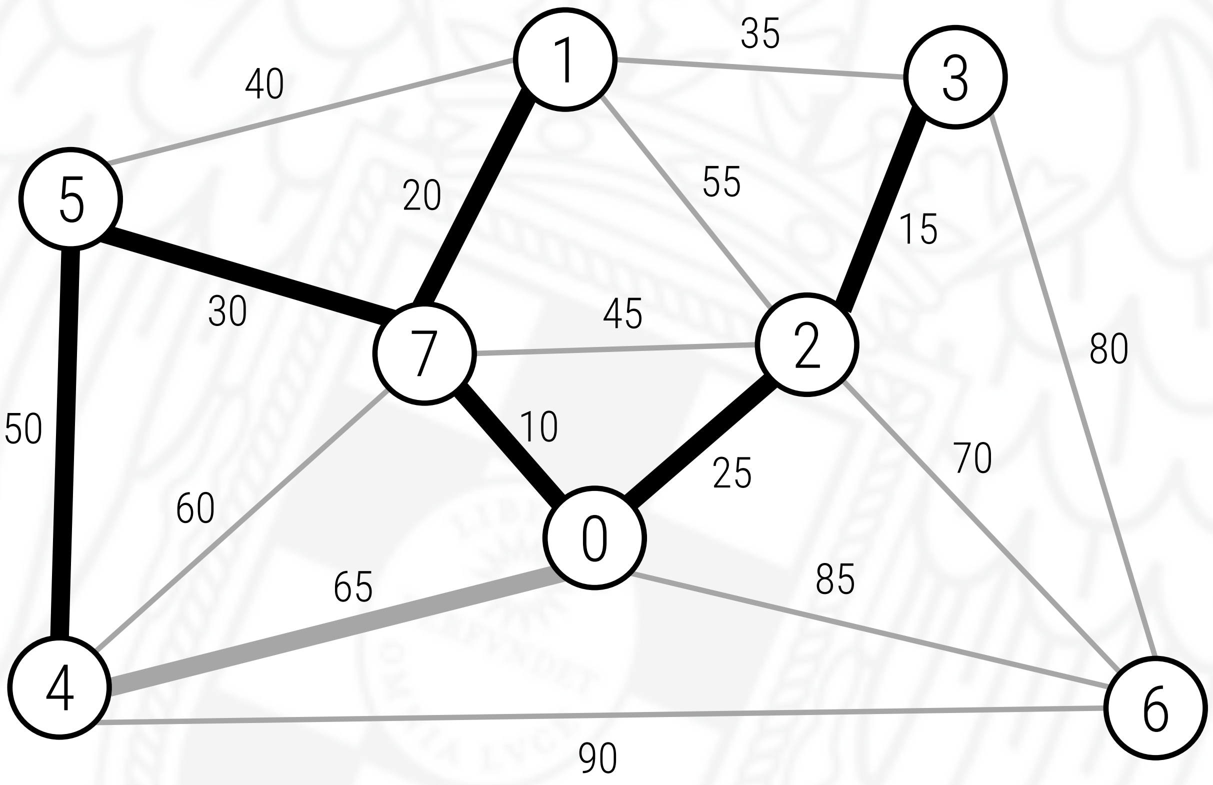
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



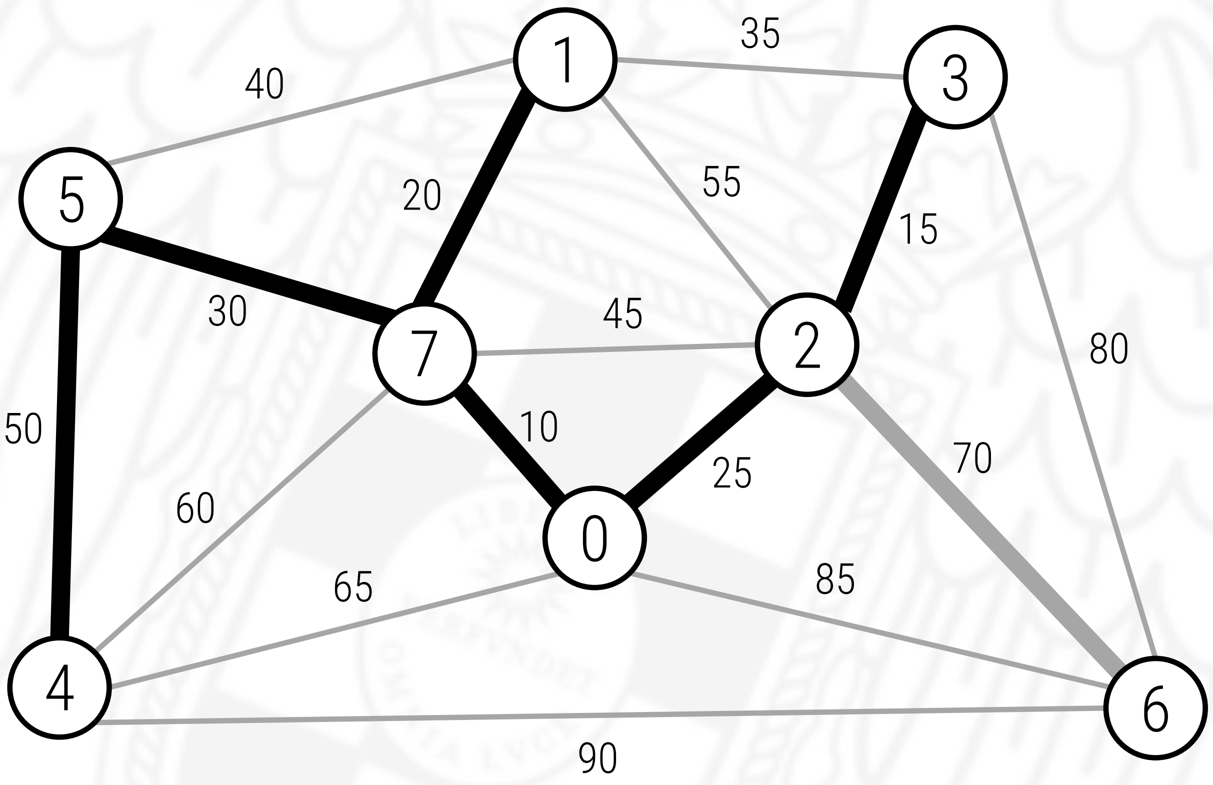
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



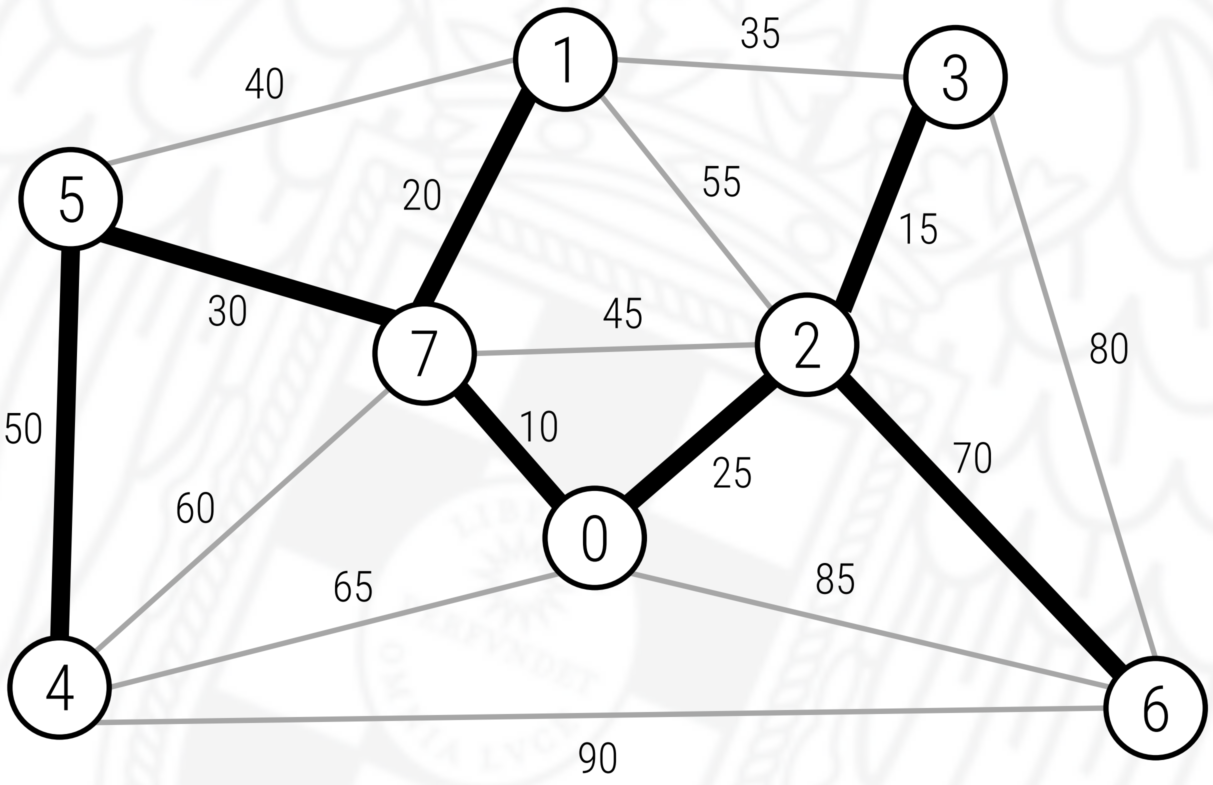
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



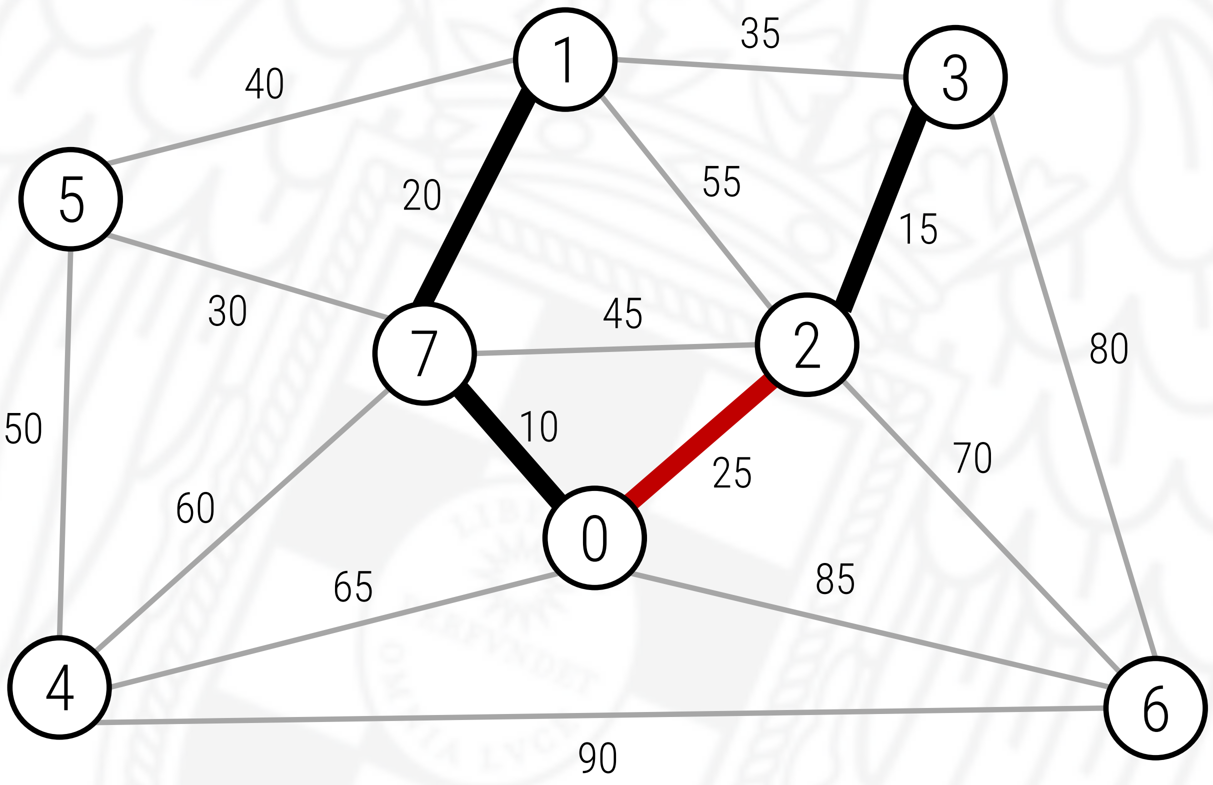
Algoritmo de Kruskal

- Considera las aristas en orden creciente de coste, y cada arista se selecciona si no crea ciclos.



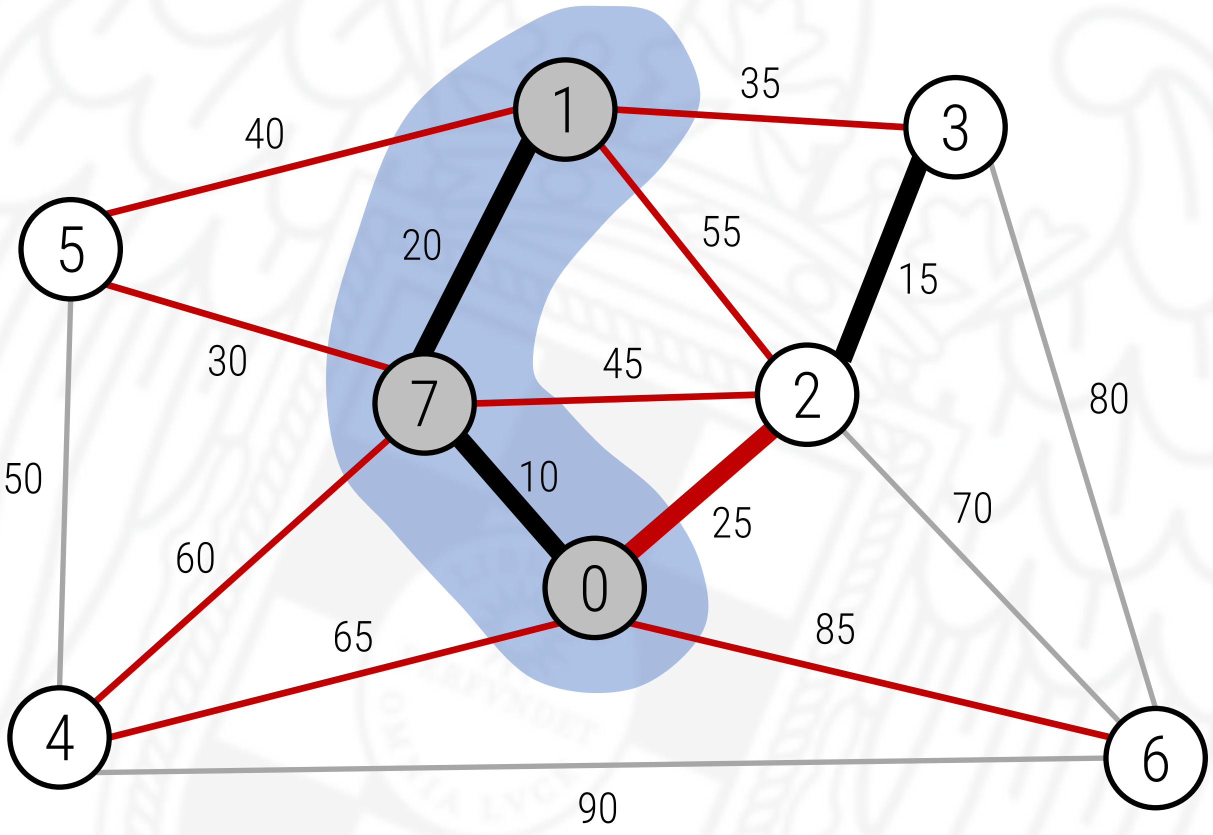
Algoritmo de Kruskal, corrección

- El algoritmo de Kruskal calcula un ARM.



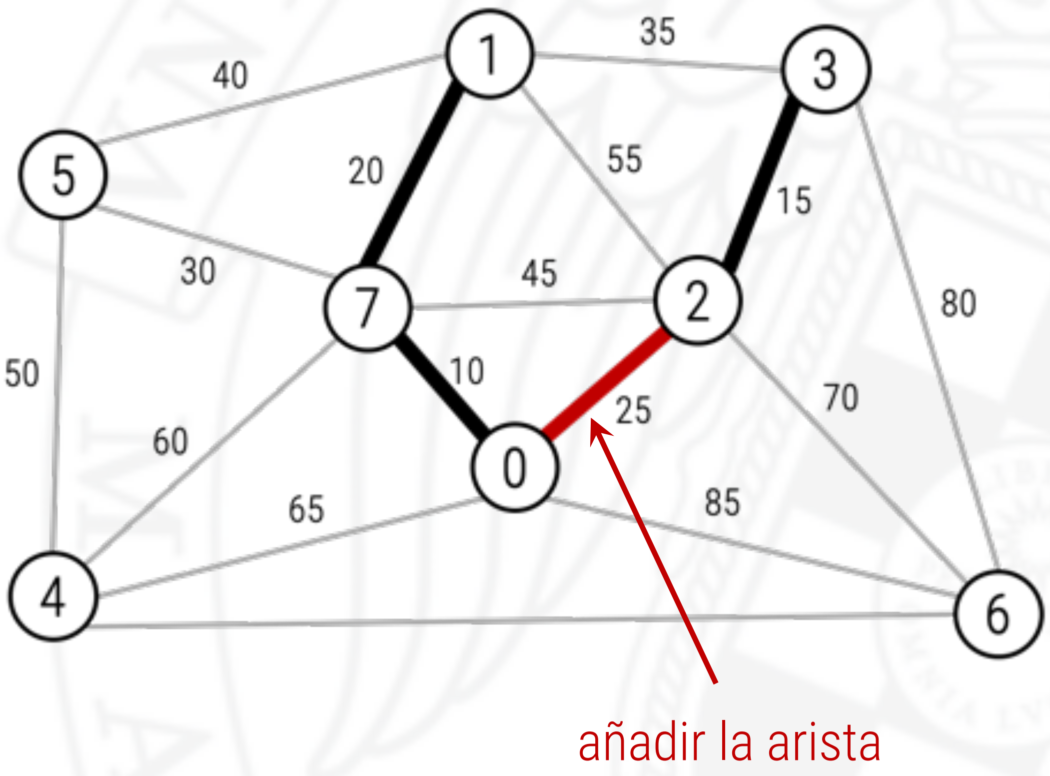
Algoritmo de Kruskal, corrección

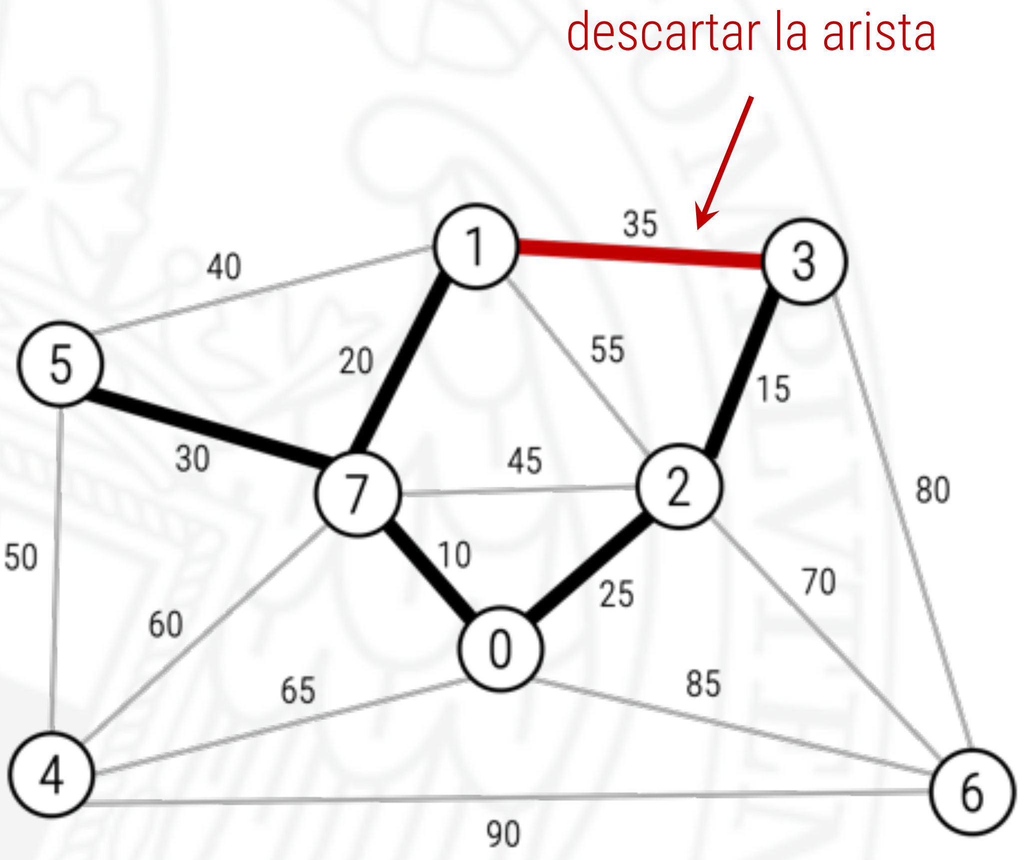
- El algoritmo de Kruskal calcula un ARM.
- Por la propiedad del corte:



Algoritmo de Kruskal, implementación

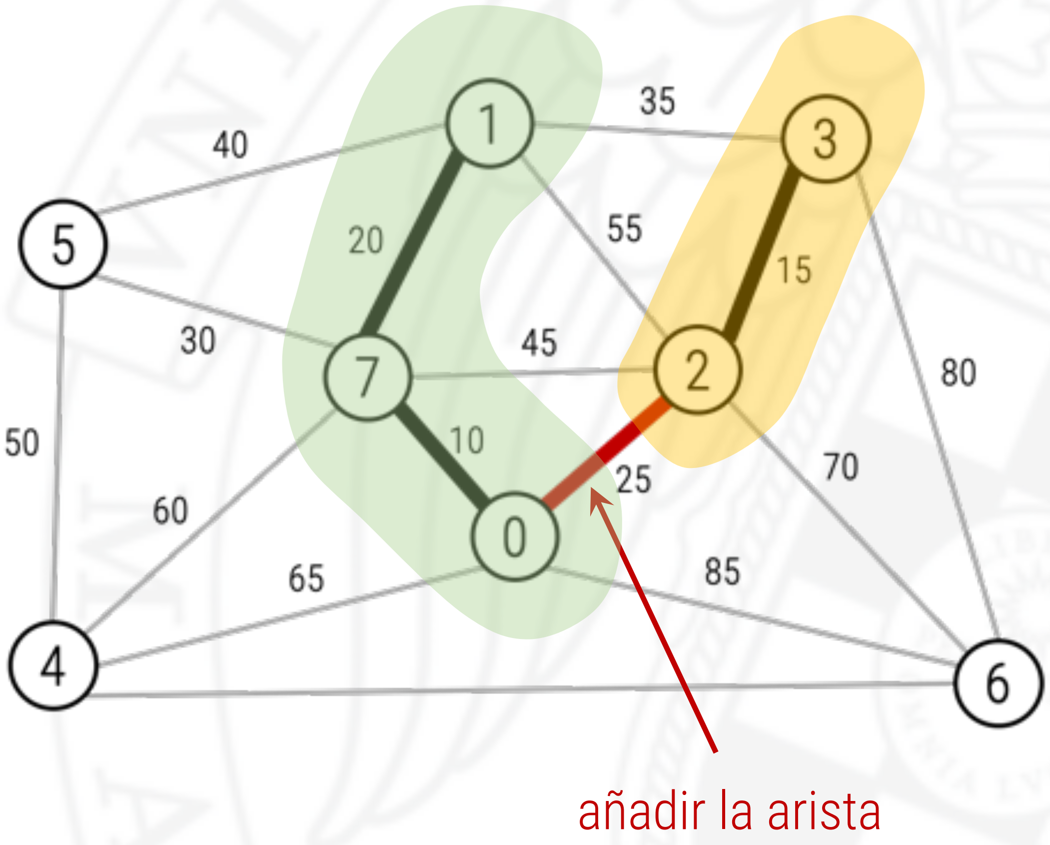
- ¿Cómo comprobamos si añadir una arista crearía ciclos?

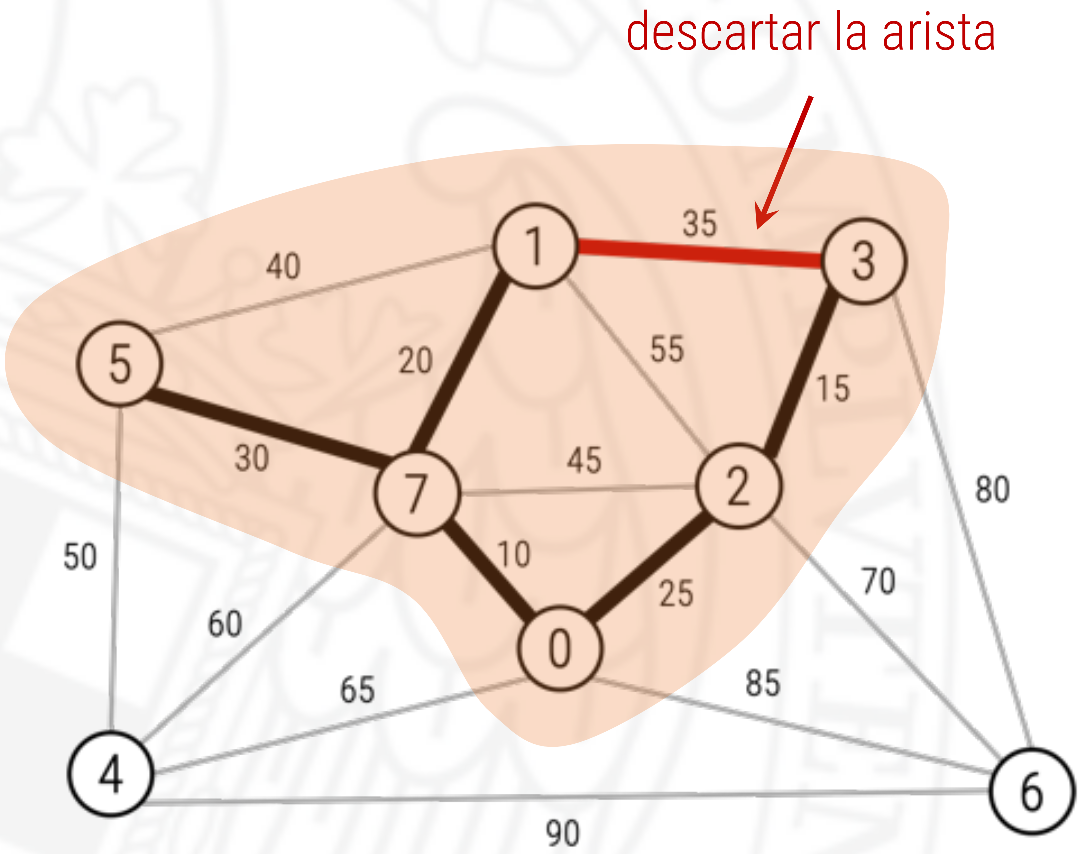




Algoritmo de Kruskal, implementación

- Utilizando conjuntos disjuntos





Algoritmo de Kruskal, implementación

```
template <typename Valor>
class ARM_Kruskal {
private:
   std::vector<Arista<Valor>> _ARM;
   Valor coste;
public:

   Valor costeARM() const {
      return coste;
   }
   std::vector<Arista<Valor>> const& ARM() const {
      return _ARM;
   }
```

Algoritmo de Kruskal, implementación

```
    ARM_Kruskal(GrafoValorado<Valor> const& g) : coste(0) {
        PriorityQueue<Arista<Valor>> pq(g.aristas());
        ConjuntosDisjuntos cjtos(g.V());
        while (!pq.empty()) {
            auto a = pq.top(); pq.pop();
            int v = a.uno(), w = a.otro(v);
            if (!cjtos.unidos(v,w)) {
                cjtos.unir(v, w);
                _ARM.push_back(a); coste += a.valor();
                if (_ARM.size() == g.V() - 1) break;
            }
        }
    }
};
```

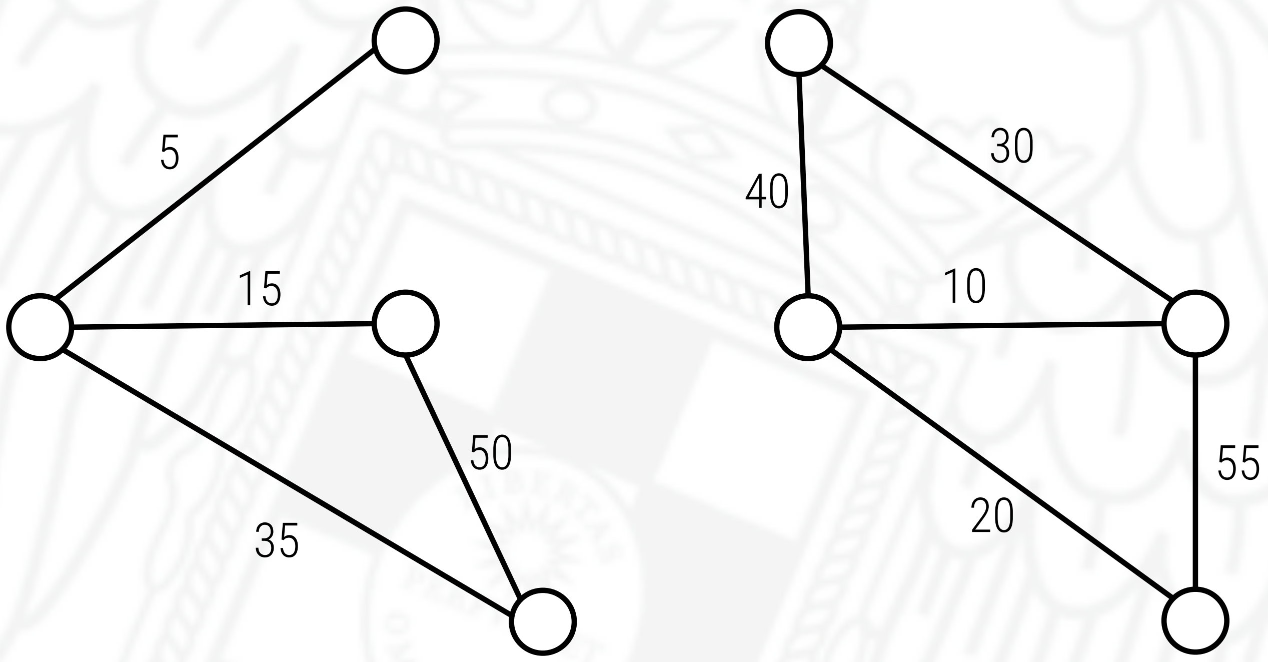
Algoritmo de Kruskal, análisis del coste

- El algoritmo de Kruskal, aplicado a un grafo con V vértices y A aristas, calcula el ARM en un tiempo en $O(A \log A)$ y con un espacio adicional en $O(A)$.

Operación	Frecuencia	Coste por operación
construir cola prioridad	1	A
construir partición	1	V
`pop`	A	$\log A$
`unir`	$V - 1$	$\lg^* V$
`unidos`	A	$\lg^* V$

Eliminación de las simplificaciones

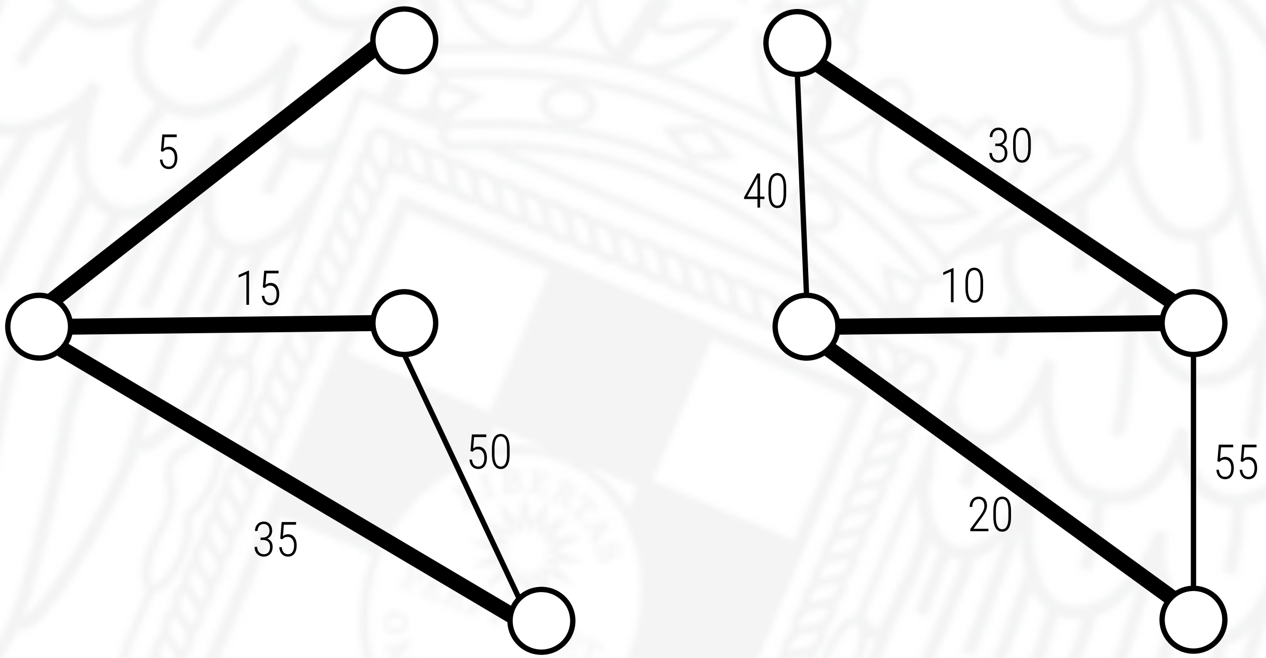
¿Qué ocurre si el grafo no es conexo?



Eliminación de las simplificaciones

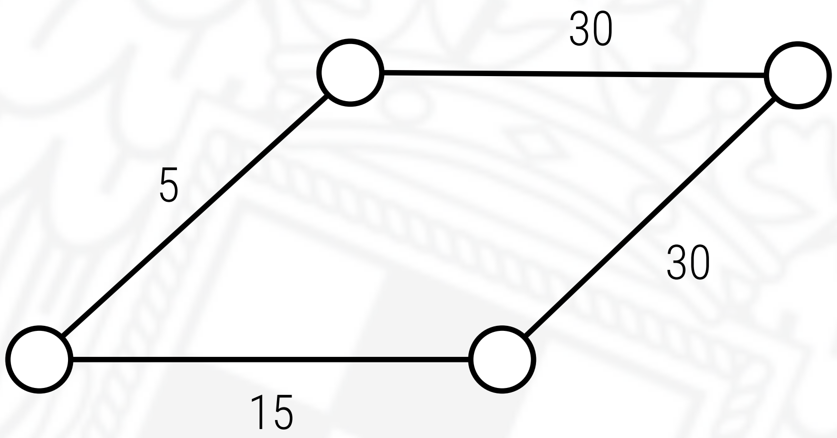
¿Qué ocurre si el grafo no es conexo?

- Se calcula un bosque de recubrimiento mínimo.



Eliminación de las simplificaciones

¿Qué ocurre si los costes de las aristas no son todos distintos?



Eliminación de las simplificaciones

¿Qué ocurre si los costes de las aristas no son todos distintos?