

Computación y Estructuras Discretas I

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes



2024-2

Agenda del día

1 **Árbol generador mínimo**

- **Árbol generador**
- Árbol generador mínimo

2 **Algoritmos para árboles generadores mínimos**

- Introducción
- Algoritmo de Prim
- Algoritmo de Kruskal
- Ejercicios

Árbol generador

¿Qué es un árbol generador?

Árbol generador

¿Qué es un árbol generador?

- Sea G un grafo simple, un árbol generador (o recubridor, o de expansión) de G es un subgrafo de G que es un árbol y contiene todos los vértices de G .

Árbol generador

¿Qué podemos decir de este grafo que admite un árbol generador?

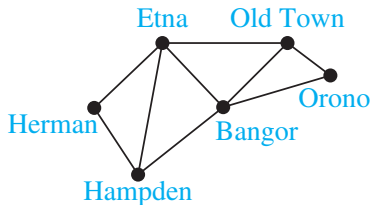
Árbol generador

¿Qué podemos decir de este grafo que admite un árbol generador?

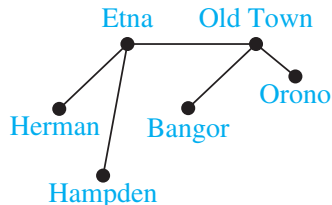
- Un grafo simple que admite un árbol generador necesariamente es conexo, puesto que existe un camino en el árbol generador entre dos vértices cualesquiera.
- Todo grafo simple conexo tiene un árbol generador.

Árbol generador

Ejemplo



(a)



(b)

Árbol generador

¿Cómo se obtiene un árbol generador?

Árbol generador

¿Cómo se obtiene un árbol generador?

- Utilizando DFS o BFS.

Agenda del día

1 Árbol generador mínimo

- Árbol generador
- Árbol generador mínimo

2 Algoritmos para árboles generadores mínimos

- Introducción
- Algoritmo de Prim
- Algoritmo de Kruskal
- Ejercicios

Árbol generador mínimo

¿Qué es un árbol generador mínimo?

Árbol generador mínimo

¿Qué es un árbol generador mínimo?

- Un árbol generador mínimo de un grafo ponderado es un árbol generador tal que la suma de los pesos de sus aristas es la mínima posible de entre todos los árboles generadores.

Árbol generador mínimo

¿Para que sirve un árbol generador mínimo?

Árbol generador mínimo

¿Para que sirve un árbol generador mínimo?

- Para resolver una amplia variedad de problemas mediante el cálculo de un árbol generador de un grafo ponderado tal que la suma de los pesos de las aristas del árbol sea mínima.
- Por ejemplo, una compañía planea construir una red de comunicaciones para conectar sus cinco centros informáticos.
- Cualquier pareja de estos centros debe estar enlazada mediante una línea telefónica alquilada.

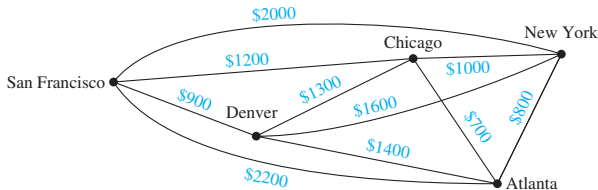
Árbol generador mínimo

¿Qué enlaces deberían realizarse para asegurar que hay un camino entre cualquier par de centros de modo que el coste total de la red sea el menor posible?

Árbol generador mínimo

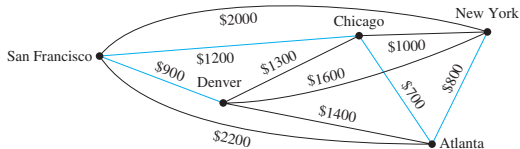
¿Qué enlaces deberían realizarse para asegurar que hay un camino entre cualquier par de centros de modo que el coste total de la red sea el menor posible?

- Podemos modelar este problema usando el siguiente grafo, donde los vértices representan los centros informáticos, las aristas representan posibles líneas de comunicaciones y los pesos de las aristas son los costes mensuales del alquiler de las líneas asociadas a las aristas.



Árbol generador mínimo

- Se puede resolver este problema encontrando un árbol generador de manera que la suma de pesos de las aristas del árbol sea mínima.
- Este árbol generador se llama árbol generador mínimo.



Agenda del día

1 **Árbol generador mínimo**

- Árbol generador
- Árbol generador mínimo

2 **Algoritmos para árboles generadores mínimos**

- **Introducción**
- Algoritmo de Prim
- Algoritmo de Kruskal
- Ejercicios

Algoritmos para árboles generadores mínimos

¿Cómo construir árboles generadores de peso mínimo?

Algoritmos para árboles generadores mínimos

¿Cómo construir árboles generadores de peso mínimo?

- Utilizando dos algoritmos muy conocidos:
 - Algoritmo de Prim.
 - Algoritmo de Kruskal.

Algoritmos para árboles generadores mínimos

¿Cómo trabajan estos algoritmos?

Algoritmos para árboles generadores mínimos

¿Cómo trabajan estos algoritmos?

- Ambos actúan añadiendo sucesivamente aristas de peso mínimo de entre todas aquellas que no han sido utilizadas y que verifican una propiedad dada.
- Ambos son algoritmos voraces.
 - Un algoritmo voraz es un procedimiento que realiza una elección óptima en cada uno de sus pasos.
 - El hecho de que se optimice en cada paso no garantiza que la solución final sea una solución óptima del problema.
- Sin embargo, estos dos algoritmos para la construcción de árboles generadores de peso mínimo son algoritmos voraces que proporcionan soluciones óptimas del problema.

Agenda del día

1 **Árbol generador mínimo**

- Árbol generador
- Árbol generador mínimo

2 **Algoritmos para árboles generadores mínimos**

- Introducción
- **Algoritmo de Prim**
- Algoritmo de Kruskal
- Ejercicios

Algoritmo de Prim

¿Cómo funciona el algoritmo de Prim?

Algoritmo de Prim

¿Cómo funciona el algoritmo de Prim?

- Para ejecutar el algoritmo de Prim, comenzamos eligiendo cualquier arista de peso mínimo y la seleccionamos para el árbol.
- Añadimos sucesivamente aristas al árbol de entre las de peso mínimo que sean incidentes con un vértice que ya está en el árbol y que no formen un ciclo con otras aristas del árbol.
- Paramos cuando hayamos añadido $n - 1$ aristas.

Algoritmo de Prim

¿Cuál es el pseudocódigo del algoritmo de Prim?

Algoritmo de Prim

¿Cuál es el pseudocódigo del algoritmo de Prim?

ALGORITHM 1 Prim's Algorithm.

procedure *Prim*(G : weighted connected undirected graph with n vertices)

T := a minimum-weight edge

for i := 1 **to** $n - 2$

e := an edge of minimum weight incident to a vertex in T and not forming a simple circuit in T if added to T

T := T with e added

return T { T is a minimum spanning tree of G }

Algoritmo de Prim

¿Qué se debe tener en cuenta a la hora de ejecutar el algoritmo?

Algoritmo de Prim

¿Qué se debe tener en cuenta a la hora de ejecutar el algoritmo?

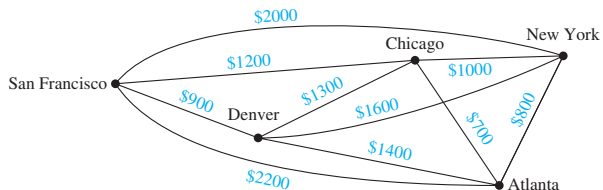
- La elección de una arista para añadirla en cualquiera de los pasos del algoritmo no está determinada cuando hay más de una con el mismo peso satisfaciendo los criterios.
- Se deben ordenar las aristas para realizar la elección de manera determinista.
- También cabe señalar que hay más de un árbol generador de peso mínimo para una grafo simple conexo y ponderado.

Algoritmo de Prim

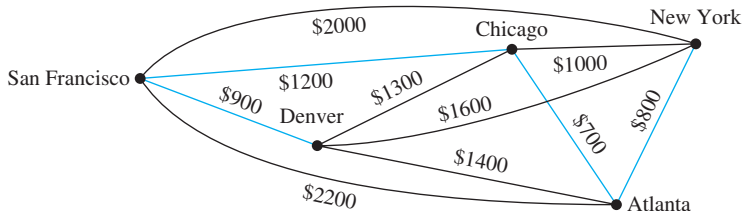
¿Cómo sería el seguimiento de este algoritmo sobre el siguiente grafo?

Algoritmo de Prim

¿Cómo sería el seguimiento de este algoritmo sobre el siguiente grafo?



Algoritmo de Prim



Choice	Edge	Cost
1	{Chicago, Atlanta}	\$ 700
2	{Atlanta, New York}	\$ 800
3	{Chicago, San Francisco}	\$1200
4	{San Francisco, Denver}	\$ 900
Total:		\$3600

Algoritmo de Prim

¿Cómo sería el pseudocódigo del algoritmo utilizando colas de prioridad?

Algoritmo de Prim

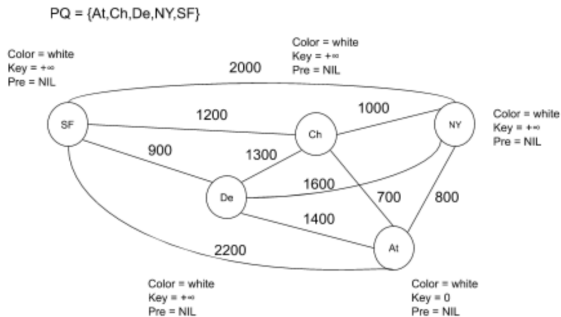
¿Cómo sería el pseudocódigo del algoritmo utilizando colas de prioridad?

```

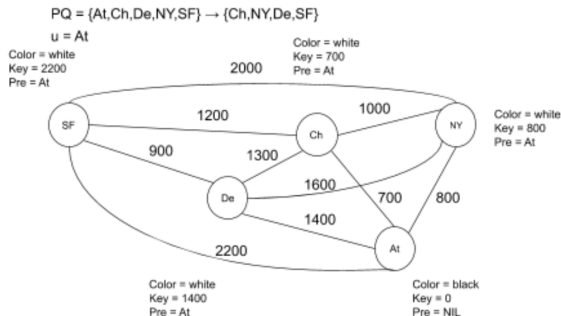
Prim( $G, w, r$ ) {
  for each  $u \in V$  {
     $key[u] := +\infty$ ;
     $color[u] := W$ ;
  }
   $key[r] := 0$ ;
   $pred[r] := NIL$ ;
   $Q = \text{new PriorityQueue}(V)$ ;
  while ( $Q$  is nonempty) {
     $u = Q.\text{extractMin}()$ ;
    for each ( $v \in adj[u]$ ) {
      if ( $color[v] = W \wedge w[u, v] < key[v]$ ) {
         $key[v] := w[u, v]$ ;
         $Q.\text{decreaseKey}(v, key[v])$ ;
         $pred[v] := u$ ;
      }
    }
     $color[u] := B$ ;
  }
}

```

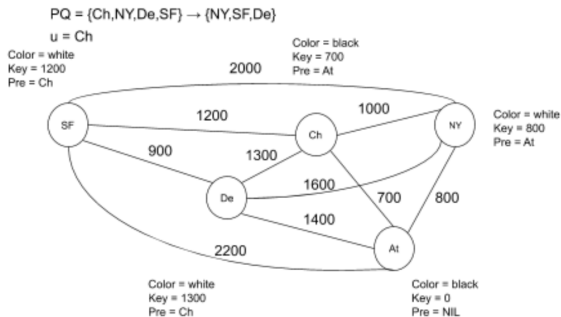
Algoritmo de Prim



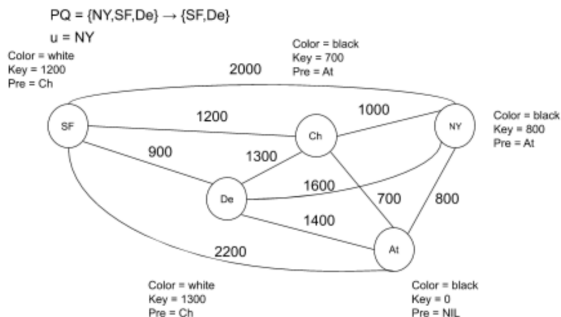
Algoritmo de Prim



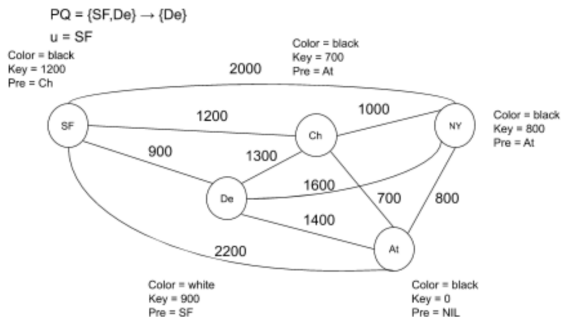
Algoritmo de Prim



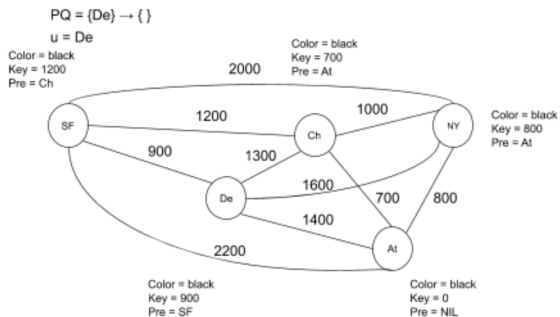
Algoritmo de Prim



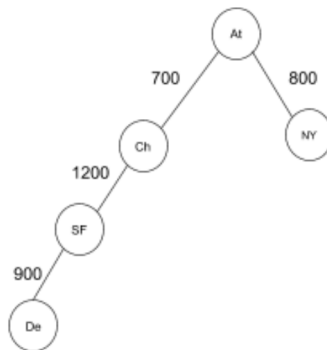
Algoritmo de Prim



Algoritmo de Prim



Algoritmo de Prim



$$\text{Total} = 700 + 800 + 1200 + 900 = 3600$$

Agenda del día

1 **Árbol generador mínimo**

- Árbol generador
- Árbol generador mínimo

2 **Algoritmos para árboles generadores mínimos**

- Introducción
- Algoritmo de Prim
- **Algoritmo de Kruskal**
- Ejercicios

Algoritmo de Kruskal

¿Cómo funciona el algoritmo de Kruskal?

Algoritmo de Kruskal

¿Cómo funciona el algoritmo de Kruskal?

- Para ejecutar el algoritmo de Kruskal, se elige una arista del grafo de entre las de menor peso.
- Se añaden paulatinamente las aristas con menor peso siempre que éstas no formen un ciclo con las otras ya incorporadas.
- El proceso termina cuando se han seleccionado $n - 1$ aristas.

Algoritmo de Kruskal

¿Cuál es el pseudocódigo del algoritmo de Kruskal?

Algoritmo de Kruskal

¿Cuál es el pseudocódigo del algoritmo de Kruskal?

ALGORITHM 2 Kruskal's Algorithm.

```
procedure Kruskal( $G$ : weighted connected undirected graph with  $n$  vertices)  
   $T :=$  empty graph  
  for  $i := 1$  to  $n - 1$   
     $e :=$  any edge in  $G$  with smallest weight that does not form a simple circuit  
      when added to  $T$   
     $T := T$  with  $e$  added  
  return  $T$  { $T$  is a minimum spanning tree of  $G$ }
```

Algoritmo de Kruskal

¿Qué se debe tener en cuenta a la hora de ejecutar el algoritmo?

Algoritmo de Kruskal

¿Qué se debe tener en cuenta a la hora de ejecutar el algoritmo?

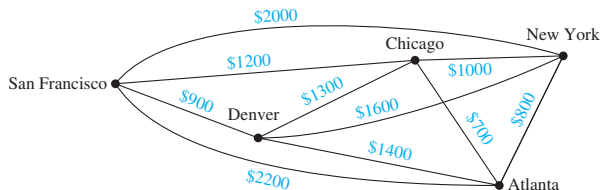
- La elección de una arista para añadirla en cualquiera de los pasos del algoritmo no está determinada cuando hay más de una con el mismo peso satisfaciendo los criterios.
- Se deben ordenar las aristas para realizar la elección de manera determinista.
- También cabe señalar que hay más de un árbol generador de peso mínimo para una grafo simple conexo y ponderado.

Algoritmo de Kruskal

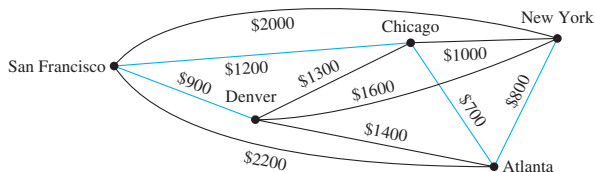
¿Cómo sería el seguimiento de este algoritmo sobre el siguiente grafo?

Algoritmo de Kruskal

¿Cómo sería el seguimiento de este algoritmo sobre el siguiente grafo?



Algoritmo de Kruskal



Elección	Arista	Coste
1	{Chicago, Atlanta}	700
2	{New York, Atlanta}	800
3	{San Francisco, Denver}	900
4	{Chicago, San Francisco}	1200

Algoritmo de Kruskal

¿Cuál es otra interpretación del funcionamiento de este algoritmo?

Algoritmo de Kruskal

¿Cuál es otra interpretación del funcionamiento de este algoritmo?

- Inicialmente se crean $|V|$ árboles de un solo nodo (o conjuntos).
- En cada iteración se toma la menor arista (u, v) restante de la lista, se toman los dos árboles (o conjuntos) que contienen a u y a v y se verifica que estos árboles o conjuntos no sean el mismo.
- Luego de todas estas verificaciones, los árboles (o conjuntos) se conectan (o mezclan).

Algoritmo de Kruskal

¿Cuál es otra interpretación del funcionamiento de este algoritmo?

- Inicialmente se crean $|V|$ árboles de un solo nodo (o conjuntos).
- En cada iteración se toma la menor arista (u, v) restante de la lista, se toman los dos árboles (o conjuntos) que contienen a u y a v y se verifica que estos árboles o conjuntos no sean el mismo.
- Luego de todas estas verificaciones, los árboles (o conjuntos) se conectan (o mezclan).

¿Por qué lo anterior garantiza que no hayan ciclos?

Algoritmo de Kruskal

¿Cuál es otra interpretación del funcionamiento de este algoritmo?

- Inicialmente se crean $|V|$ árboles de un solo nodo (o conjuntos).
- En cada iteración se toma la menor arista (u, v) restante de la lista, se toman los dos árboles (o conjuntos) que contienen a u y a v y se verifica que estos árboles o conjuntos no sean el mismo.
- Luego de todas estas verificaciones, los árboles (o conjuntos) se conectan (o mezclan).

¿Por qué lo anterior garantiza que no hayan ciclos?

- Ya que no es posible conectar dos árboles que no comparten vértices con una sola arista y crear un ciclo.

Algoritmo de Kruskal

¿Cómo implementar este algoritmo?

Algoritmo de Kruskal

¿Cómo implementar este algoritmo?

- Utilizando una nueva estructura de datos denominada conjuntos disjuntos o union-find.

Algoritmo de Kruskal

¿Cómo implementar este algoritmo?

- Utilizando una nueva estructura de datos denominada conjuntos disjuntos o union-find.

¿Qué son los conjuntos disjuntos?

Algoritmo de Kruskal

¿Cómo implementar este algoritmo?

- Utilizando una nueva estructura de datos denominada conjuntos disjuntos o union-find.

¿Qué son los conjuntos disjuntos?

- Es una estructura de datos con particiones.
- Una partición es un conjunto de conjuntos tales que cada elemento pertenece a uno y sólo un conjunto.

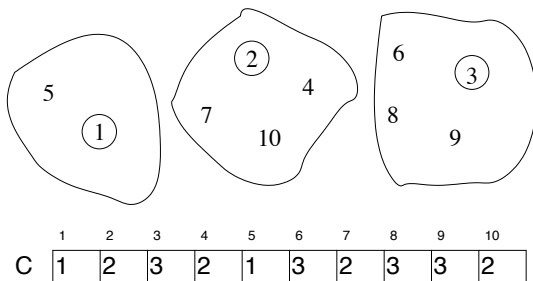
Algoritmo de Kruskal

¿Cómo se representan estos conjuntos disjuntos?

Algoritmo de Kruskal

¿Cómo se representan estos conjuntos disjuntos?

- Todos los elementos se numeran de 1 a n .
- Cada subconjunto toma su nombre de uno de sus elementos, su representante, ej. el valor más pequeño.
- Se mantiene en un vector el nombre del subconjunto disjunto de cada elemento



Algoritmo de Kruskal

¿Cuáles son las operaciones de los conjuntos disjuntos?

Algoritmo de Kruskal

¿Cuáles son las operaciones de los conjuntos disjuntos?

- `makeSet(x)` - Crea un conjunto de un solo elemento.
- `find(x)` - Encuentra el conjunto al que pertenece x .
- `union(x, y)` - Realiza la unión de los conjuntos que contienen a x y a y .

Se asume que los elementos se representan como enteros, que pueden ser los índices de un arreglo.

Algoritmo de Kruskal

¿Cómo se implementan los conjuntos disjuntos?

Algoritmo de Kruskal

¿Cómo se implementan los conjuntos disjuntos?

- Existen dos implementaciones populares: listas enlazadas y árboles.

Algoritmo de Kruskal

¿Cómo se implementan los conjuntos disjuntos?

- Existen dos implementaciones populares: listas enlazadas y árboles.

¿Cómo hacerlo con listas enlazadas?

Algoritmo de Kruskal

¿Cómo se implementan los conjuntos disjuntos?

- Existen dos implementaciones populares: listas enlazadas y árboles.

¿Cómo hacerlo con listas enlazadas?

- Se utilizan listas enlazadas para representar los conjuntos y un arreglo, denominado arreglo de representantes.
- El arreglo se encuentra indizado por el número del elemento y el valor lo da el nombre de dicho conjunto (el menor miembro del conjunto).
- La operación makeset resulta obvia, crear el arreglo de representantes y las listas enlazadas de un sólo elemento cada una.
- La operación find tan sólo implica acceder al arreglo de representantes.
- La unión mezcla las listas y se debe actualizar el arreglo de representantes.

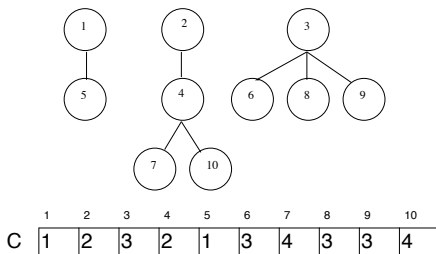
Algoritmo de Kruskal

¿Cómo hacerlo con árboles?

Algoritmo de Kruskal

¿Cómo hacerlo con árboles?

- Se utiliza un árbol para caracterizar cada subconjunto.
- La raíz nombra al subconjunto.
- La representación de los árboles es fácil porque la única información necesaria es un apuntador al padre.
- Cada entrada $p[i]$ en el vector contiene el padre del elemento i .
- Si i es una raíz, entonces $p[i] = i$



Algoritmo de Kruskal

¿Cómo se implementan las operaciones?

Algoritmo de Kruskal

¿Cómo se implementan las operaciones?

- Una búsqueda sobre el elemento x se efectúa devolviendo la raíz del árbol que contiene x .
- La unión de dos conjuntos se efectúa combinando ambos árboles: apuntando la raíz de un árbol a la del otro.

Algoritmo de Kruskal

¿Qué es la unión por alturas?

Algoritmo de Kruskal

¿Qué es la unión por alturas?

- Las uniones anteriores se efectuaban de modo arbitrario.
- Una mejora sencilla es realizar las uniones haciendo del árbol menos profundo un subárbol del árbol más profundo.
- La altura se incrementa sólo cuando se unen dos árboles de igual altura.

Algoritmo de Kruskal

¿Cómo sería el pseudocódigo?

Algoritmo de Kruskal

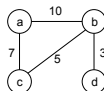
¿Cómo sería el pseudocódigo?

```
KRUSKAL(G):  
1  A =  $\emptyset$   
2  foreach v  $\in$  G.V:  
3      MAKE-SET(v)  
4  foreach (u, v) in G.E ordered by weight(u, v), increasing:  
5      if FIND-SET(u)  $\neq$  FIND-SET(v):  
6          A = A  $\cup$  {(u, v)}  
7          UNION(u, v)  
8  return A
```

Ejemplo

Ejemplo

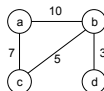
¿Cómo sería el seguimiento al pseudocódigo previo con el siguiente grafo?



Ejemplo

Ejemplo

¿Cómo sería el seguimiento al pseudocódigo previo con el siguiente grafo?



$$A = \emptyset$$

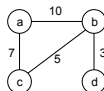
$$PQ = \{(b,d), (b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b\}, \{c\}, \{d\}\}$$

Ejemplo

Ejemplo

¿Cómo sería el seguimiento al pseudocódigo previo con el siguiente grafo?



$$A = \emptyset$$

$$PQ = \{(b,d), (b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b\}, \{c\}, \{d\}\}$$

$$A = \{(b,d)\}$$

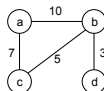
$$PQ = \{(b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b,d\}, \{c\}\}$$

Ejemplo

Ejemplo

¿Cómo sería el seguimiento al pseudocódigo previo con el siguiente grafo?



$$A = \emptyset$$

$$PQ = \{(b,d), (b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b\}, \{c\}, \{d\}\}$$

$$A = \{(b,d)\}$$

$$PQ = \{(b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b,d\}, \{c\}\}$$

$$A = \{(b,d), (b,c)\}$$

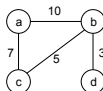
$$PQ = \{(a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b,c,d\}\}$$

Ejemplo

Ejemplo

¿Cómo sería el seguimiento al pseudocódigo previo con el siguiente grafo?



$A = \emptyset$

$PQ = \{(b,d), (b,c), (a,c), (a,b)\}$

$DS = \{\{a\}, \{b\}, \{c\}, \{d\}\}$

$A = \{(b,d), (b,c)\}$

$PQ = \{(a,c), (a,b)\}$

$DS = \{\{a\}, \{b,c,d\}\}$

$A = \{(b,d)\}$

$PQ = \{(b,c), (a,c), (a,b)\}$

$DS = \{\{a\}, \{b,d\}, \{c\}\}$

$A = \{(b,d), (b,c), (a,c)\}$

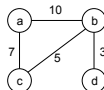
$PQ = \{(a,b)\}$

$DS = \{\{a,b,c,d\}\}$

Ejemplo

Ejemplo

¿Cómo sería el seguimiento al pseudocódigo previo con el siguiente grafo?



$$A = \emptyset$$

$$PQ = \{(b,d), (b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b\}, \{c\}, \{d\}\}$$

$$A = \{(b,d), (b,c)\}$$

$$PQ = \{(a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b,c,d\}\}$$

$$A = \{(b,d), (b,c), (a,c)\}$$

$$PQ = \emptyset$$

$$DS = \{\{a,b,c,d\}\}$$

$$A = \{(b,d)\}$$

$$PQ = \{(b,c), (a,c), (a,b)\}$$

$$DS = \{\{a\}, \{b,d\}, \{c\}\}$$

$$A = \{(b,d), (b,c), (a,c)\}$$

$$PQ = \{(a,b)\}$$

$$DS = \{\{a,b,c,d\}\}$$

Ejemplo

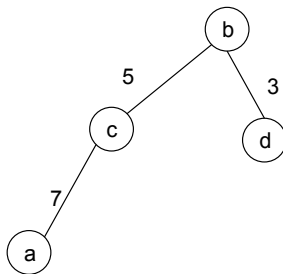
Ejemplo

¿Cómo sería el árbol de recubrimiento mínimo?

Ejemplo

Ejemplo

¿Cómo sería el árbol de recubrimiento mínimo?



Agenda del día

1 **Árbol generador mínimo**

- Árbol generador
- Árbol generador mínimo

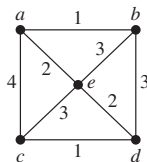
2 **Algoritmos para árboles generadores mínimos**

- Introducción
- Algoritmo de Prim
- Algoritmo de Kruskal
- **Ejercicios**

Ejercicios

Ejercicio

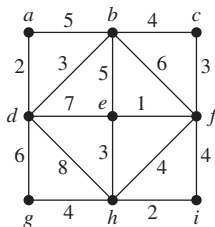
Realice el seguimiento al algoritmo de Prim y Kruskal sobre este grafo.



Ejercicios

Ejercicio

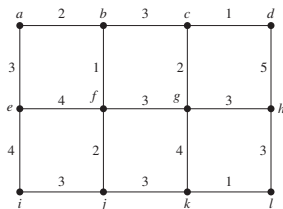
Realice el seguimiento al algoritmo de Prim y Kruskal sobre este grafo.



Ejercicios

Ejercicio

Realice el seguimiento al algoritmo de Prim y Kruskal sobre este grafo.



Ejercicios

Ejercicio

Continúe su implementación genérica de grafos (matriz de adyacencia y listas de adyacencia) e implemente el algoritmo de Prim y Kruskal (si no lo ha hecho, incluya Dijkstra, Floyd-Warshall, BFS y DFS). Recuerde que la interface debe incluir todos los métodos del grafo.