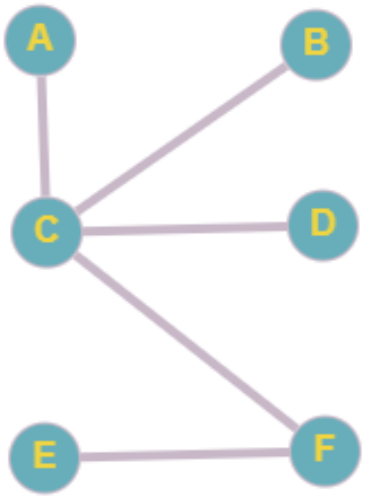
Árboles

Definiciones

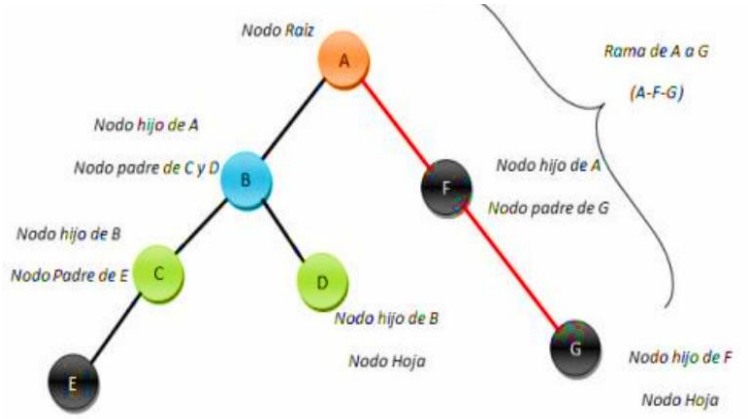
• Un **árbol** es un grafo no dirigido, conexo y sin ciclos. La siguiente figura muestra un ejemplo de árbol, donde se observa la conexión de todos los vértices a través de aristas no dirigidas y sin formación de ciclos



Es importante considerar que **un árbol de n vértices tiene n – 1 aristas.**

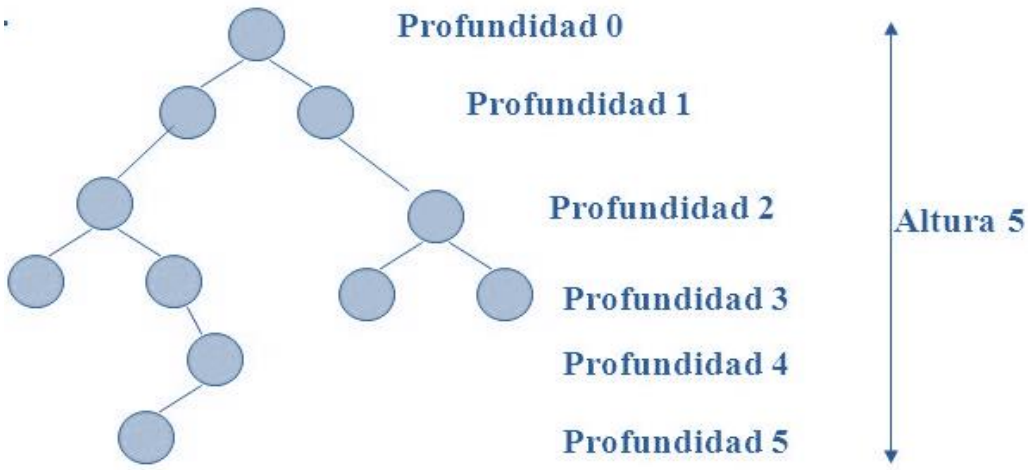
Si bien un árbol general es un grafo no dirigido, se pueden pensar una raíz especifica y una dirección implícita, y definir los siguientes conceptos:

- **Padre**: de v es el único vértice u tal que hay una arista dirigida de u a v.
- **Hijo**: cuando u es el padre de v, se dice que v es el hijo de u.
- **Hermano**: son los vértices con el mismo padre.
- **Antecesores**: de un vértice diferente de la raíz son todos los vértices en el camino desde la raíz hasta ese vértice.
- **Descendientes**: de un vértice v son aquellos vértices para los que v es un antecesor.
- **Hoja**: es un vértice que no tiene hijos.
- **Vértice interno**: es aquel que tiene hijos.
- **Subárbol**: si a es un vértice de un árbol, el subárbol con raíz en a es un subgrafo del árbol que contiene al vértice a, a todos sus descendientes y a todas las aristas incidentes en dichos descendientes



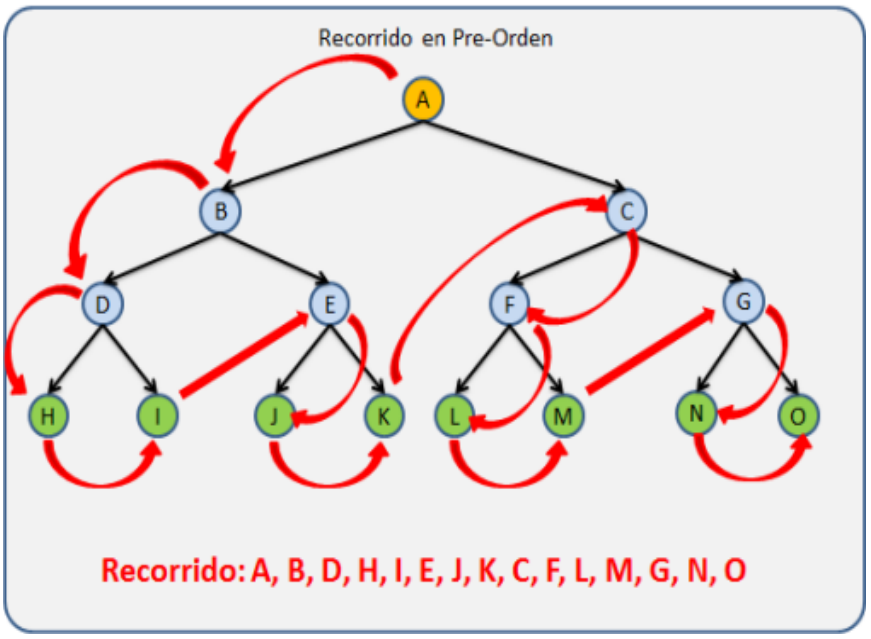
Recorridos en Árboles

- **Altura del árbol:** la altura de un árbol es el largo del mayor camino de la raíz a una hoja.

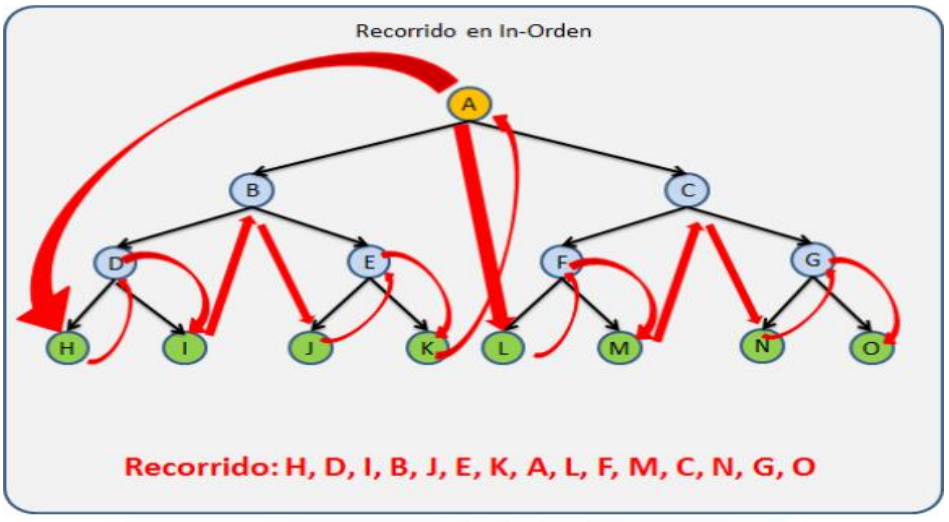


Algoritmos de recorrido

- **Recorrido en preorden**: Sea T un árbol ordenado con raíz r. Si T consta sólo de r, entonces r es el recorrido en preorden de T. En otro caso, supongamos que T_1 , T_2 , ... , T_n son subárboles de r listados de izquierda a derecha en T. El recorrido en preorden comienza visitando r, continúa recorriendo T_1, en preorden, luego T_2 y así sucesivamente hasta recorrer T_n en preorden.

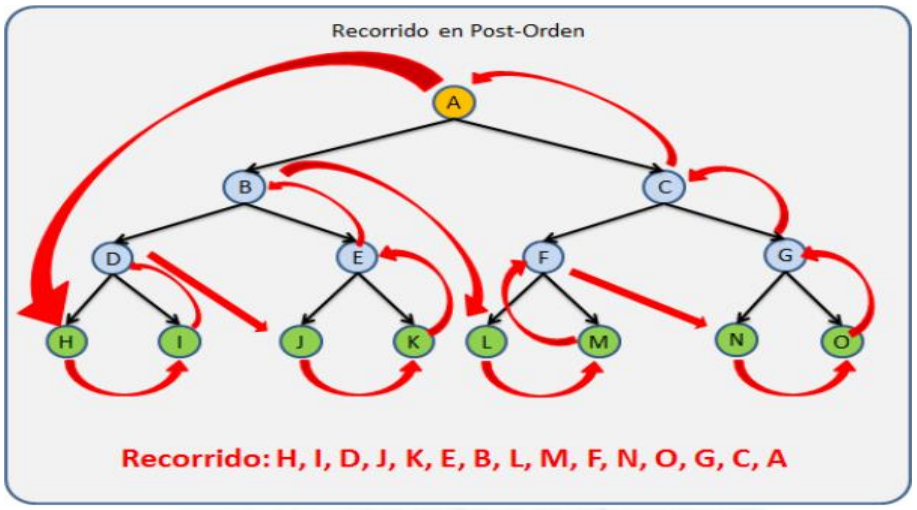


● **Recorrido en inorden:** Sea T un árbol ordenado con raíz r. Si T consta sólo de r, entonces r es el recorrido en inorden de T. En otro caso, supongamos que T_1 , T_2 , ... , T_n son subárboles de r listado de izquierda a derecha en T. El recorrido en inorden comienza recorriendo T_1 en inorden y continúa visitando r, a continuación recorre T_2 en inorden y



así sucesivamente hasta recorrer T_n en inorden.

- **Recorrido en postorden:** Sea T un árbol ordenado con raíz r. Si T consta sólo de r, entonces r es el recorrido en postorden de T. En caso, supongamos que otro T_1 , T_2 , ... , T_n son subárboles de r listado de izquierda a derecha en T. El recorrido en postorden comienza recorriendo T_1 , en postorden, luego recorre T_2 en postorden y así sucesivamente hasta recorrer Tn y finaliza visitando r.



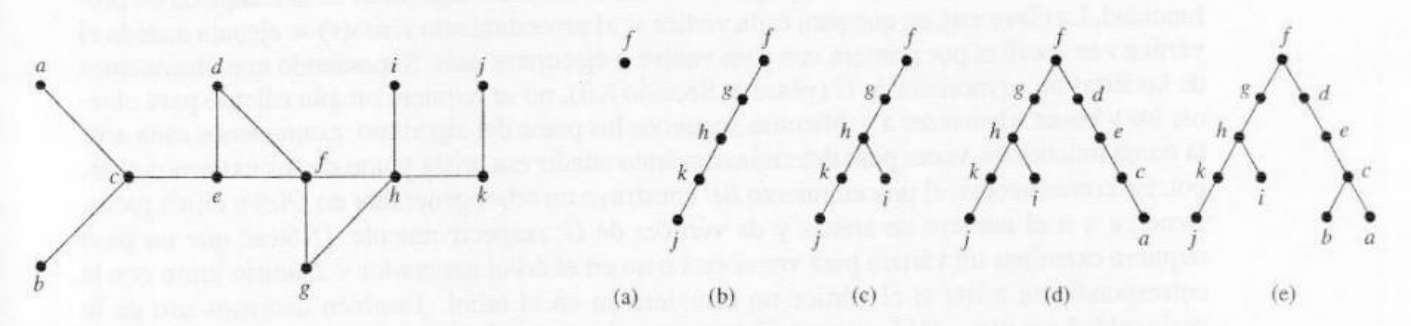
Árbol generador

- Sea G un grafo simple. Un **árbol generador** (o recubridor) de G es un subgrafo de G que es un árbol y contiene todos los vértices de G. Nótese que un grafo simple que admite un árbol generador necesariamente es conexo. También, todo grafo simple conexo tiene un árbol generado.

- **Teorema**: Un grafo simple es conexo si, y sólo si, admite un árbol generador.

- **Búsqueda en profundidad:** Podemos construir un árbol generador para un grafo simple conexo. Construiremos un árbol con raíz, y el árbol generador será el grafo no dirigido subyacente:
  - Elegimos un vértice arbitrario como raíz del árbol.
  - Formamos un camino que comienza en este vértice añadiendo sucesivamente vértices y aristas, siendo cada arista incidente con el último vértice del camino y un vértice que no está en el camino.
  - Añadimos a este camino tantos vértices y aristas como sea posible.
  - Si el camino pasa por todos los vértices, el árbol generador es dicho camino.
  - Si no, se deben añadir más vértices y aristas.

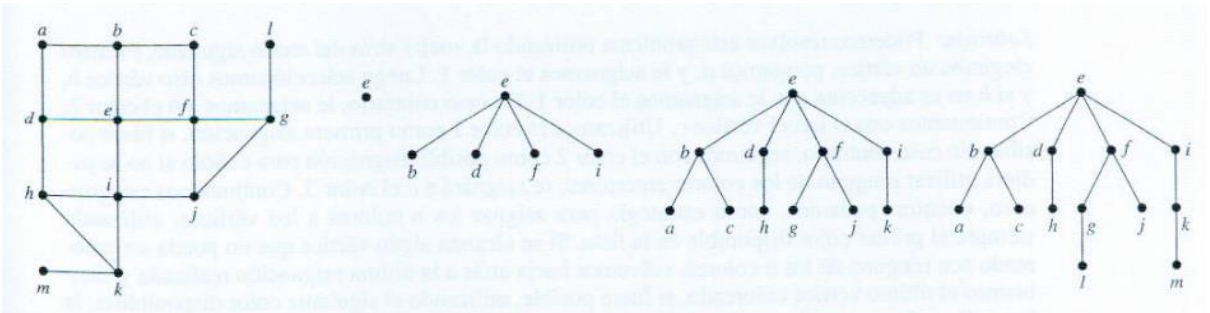
- Retrocedemos al penúltimo vértice del camino y, si es posible, formamos un nuevo camino comenzando en este vértice y que pase por los nodos no visitados. Si esto no se puede hacer, retrocedemos al vértice anterior en el recorrido hacia la raíz y lo intentamos de nuevo.
- Repetimos el proceso, hasta que no se pueda añadir más aristas.



- **Búsqueda por anchura:** Podemos construir un árbol generador para un grafo simple conexo: Construiremos un árbol con raíz, y el grafo no dirigido subyacente es el árbol generador:

- Elegimos un vértice arbitrario como raíz del árbol.
- Añadimos todas las aristas incidentes en ese vértice. Los nuevos vértices añadidos en esa fase forman los vértices del nivel 1 del árbol generador. Los ordenamos con un orden cualquiera.
- Para cada vértice del nivel 1 visitado en orden, añadimos todos los vértices incidentes con él, siempre que no formen un ciclo.
- Ordenamos los hijos de los vértices del nivel 1 con un orden cualquiera, generando así los vértices de nivel 2 del árbol.
- Repetimos este procedimiento hasta que se hayan añadido todos los vértices del árbol.

-



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.

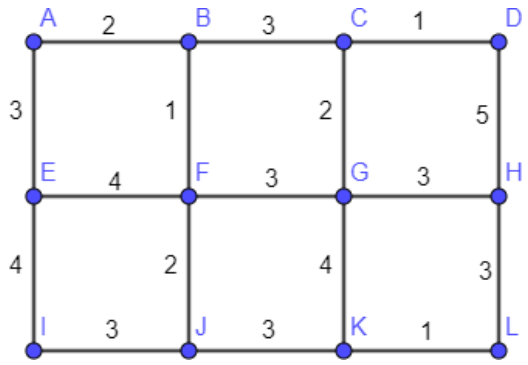
Presentaremos dos algoritmos para construir árboles generadores de pesos mínimos:

- **Algoritmo de Prim:**

- Se elige cualquier arista de peso mínimo y se la selecciona para el árbol.

- Se añaden 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 ciclos con otras aristas del árbol.

- Finaliza cuando se hayan añadido $n - 1$ aristas.



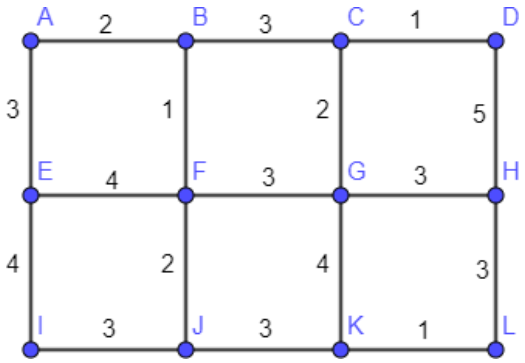
Elección	Arista	Peso
1	*b,f*	1
2	*a,b*	2
3	*f,j*	2
4	*a,e*	3
5	*i,j*	3
6	*f,g*	3
7	*c,g*	2
8	*c,d*	1
9	*g,h*	3
10	*h, l*	3
11	*k,l*	1
	TOTAL	24

- **Algoritmo de Kruskal:**

- Se elige cualquier arista de peso mínimo y se la selecciona para el árbol.

- Se añaden sucesivamente aristas al árbol de entre las de peso mínimo siempre que estas no formen un ciclo con las otras ya incorporadas. (no es necesario que sean incidentes con vértices del árbol).

- Finaliza cuando se hayan añadido $n - 1$ aristas



Elección	Arista	Peso
1	*c,d*	1
2	*k,l*	1
3	*b,f*	1
4	*c,g*	2
5	*a,b*	2
6	*f,j*	2
7	*b,c*	3
8	*j,k*	3
9	*g,h*	3
10	*i,j*	3
11	*a,e*	3
	TOTAL	24