**Estructuras Discretas**

# *Teoría de Grafos y Árboles. (3)*

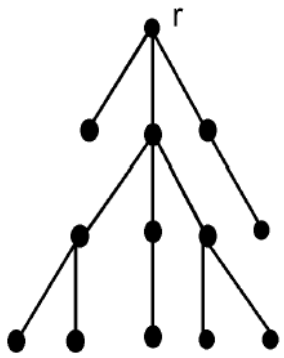
Prof. Miguel Fagúndez

## ***Árboles.***

Especialmente útil en lo que aplicaciones informáticas se refiere es un cierto tipo de grafo simple, llamado árbol, que se emplea entre otras cosas, para construir algoritmos eficientes destinados a localizar items en una lista, para construir redes de ordenadores con el mínimo costo, para construir códigos eficientes destinados a almacenar transmitir datos, para analizar algoritmos de ordenación, etc.

*Definición*: Un **árbol** es un grafo no dirigido, conexo y sin circuitos. Como un árbol no tiene circuitos, tampoco puede tener arcos múltiples ni bucles, por lo que cualquier árbol es un grafo simple.

*Definición*: Dado un árbol con raíz (G , r) se denominan **hojas** a los vértices de G distintos de r que tienen grado 1.




Es fácil darse cuenta de que las hojas de un árbol son los vértices que no tienen descendientes. A los vértices distintos de la raíz que no son hojas de un árbol se les denomina **vértices internos**.

*Definición*: Se denomina **nivel** o **profundidad** de un vértice en un árbol con raíz a la longitud del camino que une a la raíz con dicho vértice. La **altura** de un árbol con raíz es el mayor de los niveles de sus vértices (camino mas largo entre la raíz y el vértice del árbol). Ver el grafico anterior.

*Definición*: Si *a* es un vértice de un árbol con raíz (T , r), el subárbol de T que contiene a *a* como raíz es el subgrafo del árbol formado por el vértice *a*, todos sus descendientes y todas las aristas o arcos incidentes con sus descendientes.

*Definición*: Se dice que un arbol con raiz es **m_ario** si todos los vértices tienen a los sumo *m* hijos. Si todos los vértices internos (todos excepto las hojas) de un arbol m_ario tienen exactamente *m* hijos, se dice que el arbol es un arbol **m_ario completo**. A los arboles 2-arios se les denomina **arboles binarios**.

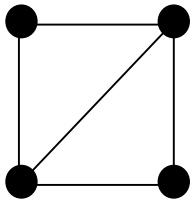
De la definición anterior podemos decir que en un árbol m_ario de altura *h* hay a lo sumo $m^h$ hojas. Y que nos lleva al siguiente colorario:

***Corolario***: Si un árbol m_ario de altura *h* tiene *l* hojas, entonces $h \geq \log_m(l)$. **Demostrar!!!**

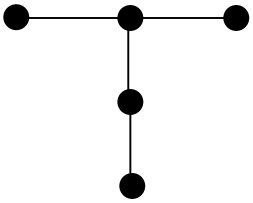
## ***Características de los Árboles*.**

1. Un árbol es un grafo conexo.
2. Un árbol no debe poseer circuitos.
3. Ningún vértice puede tener mas de un camino.

***Ejemplo***:



Esto es un Grafo pero NO un Árbol.



Esto es un Grafo y un Árbol.

**En resumen** podemos decir que un árbol es un grafo simple no dirigido que es a su vez conexo y **acíclico** (no posee circuitos). De manera formal:

Sea G = (V,A) es un árbol si y solo si:

- G es un grafo conexo, es decir, $p = 1$.
- G no posee circuitos, $P(G) = 0$.

***Árbol Expandido o de Expansión***:

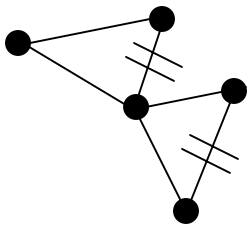
*Definición*: Es un árbol con el conjunto de vértices de V (lo cual forma un subgrafo de G); con esto es, un **árbol de expansión** es conexo acíclico y tiene todos los vértices de V y parte de los arcos de A como conjunto de arcos. Formalmente podemos decir:

Sea G = (V,A) un grafo, entonces T = ($V_1$,$A_1$) es un árbol expandido de G, si:

- T es subgrafo expandido de G ($V_1$ = V y $A_1 \subseteq A$)
- T es un árbol.

*Corolario del árbol expandido*:

Todo grafo conexo G = (V,A) posee al menos un arbol expandido T =(V,A'). Ejemplo:



G = (V,A) conexo

Dependiendo de los arcos que se extraigan de G, se obtienen diferentes árboles expandidos.

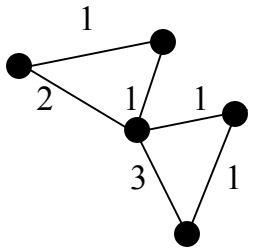
***Árbol Expandido (Cobertor) Mínimo: A.C.M.***:

El árbol expandido mínimo se estudia para grafos conexos que sean pesados. Cuyos pesos sean números reales positivos.

Definición: Sea G = (V,A) un grafo conexo y sea C: A $\rightarrow$ $R^+ \cup \{0\}$ una función que asocia a cada arco del grafo un costo, entonces C se denomina función de costo o peso de G, y podemos usar la notación: G = (V,A;C).

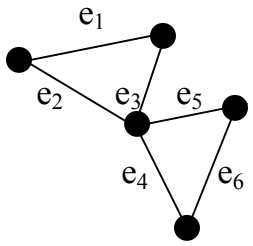
C cumple el mismo objetivo que la función de peso de la segunda definición de grafo; pero en este caso solo se considera la asociación con etiquetas correspondientes a números reales positivos incluyendo el cero.

Utilizando el grafo presentado anteriormente, veamos un ejemplo:




G = (V,A;C) conexo

Convenientemente, vamos a asociarle nombres a los arcos:

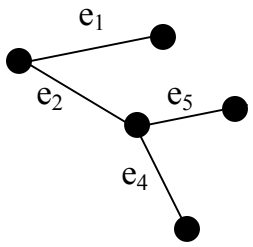



G = (V,A;C) conexo

Ahora bien, si T = (V,A;C) es un árbol expandido de G, se define el costo de T como $|A'|$ y podemos entonces decir que:

$\sum_{i=1}^{|A'|} C$ (arco $i$ de T) {*suma de los costos de todos los arcos que participan en el árbol*}

Para el grafo del ejemplo, podemos definir un árbol expandido T, de manera tal que:
T1= (V,A';C)




$C(T1) = C(e_1)+C(e_2)+C(e_4)+C(e_5) = 7$

En base a todo lo anterior podemos definir árbol expandido mínimo.

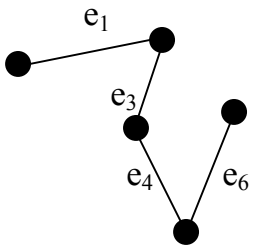
*Definición*: Un árbol expandido mínimo es un árbol expandido donde la suma de los costos de sus arcos es la menor entre la de todos los árboles expandidos de G.

*Problema de la mínima conexión*: Hallar el árbol expandido mínimo a un grafo conexo G. Dado un grafo conexo G = (V,A;C), deseamos obtener de G, un grafo $T_{min}$ = (V,A';C), tal que:

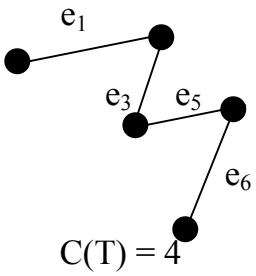
(i) $T_{min}$ sea un árbol expandido de G.

(ii) $CT_{min} \leq C_r \ \forall T$ que sea árbol expandido de G.

Otros ejemplos del grafo en estudio son:



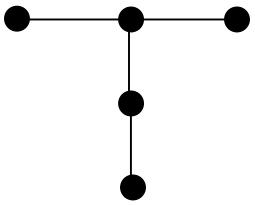

C(T) = 6

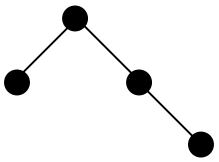



C(T) = 4

***Foresta o Bosque***:

*Definición*: Un grafo acíclico sea o no conexo, con frecuencia se denomina **Bosque**. Es claro que las componentes conexas de un bosque son árboles.





Esto es un Bosque!!

En cualquier grafo que sea árbol el numero de vértices le quitamos 1 y nos da el numero de arcos, lo que nos lleva a enunciar el siguiente teorema:

***Teorema***: Se G=(V,A) un arbol conexo y acíclico, si $|V|=n$, entonces $|A|=n-1$.

***Teorema de Caracterización Múltiple.***

Las siguientes sentencias son equivalentes para un grafo con $n \geq 2$.

1. G es un Árbol.
2. G no posee circuitos y posee n-1 arcos.
3. G es conexo y posee n-1 arcos.
4. G es arcomaximal respecto a no poseer circuitos.
5. G es arcominimal respecto a ser conexo.
6. Cada par de vértices esta unido por un único camino.

**Demostrar!!!**