



UTN-FRBA

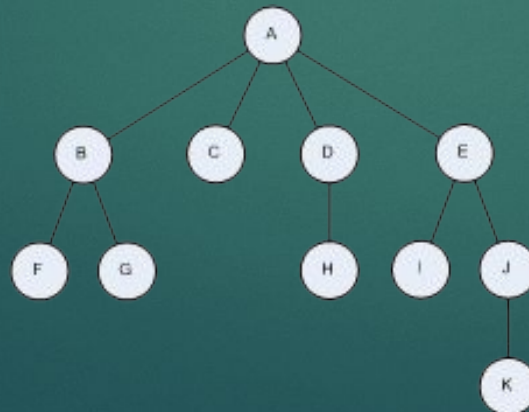
GESTION DE DATOS

ARBOLES

DIRECTOR CATEDRA: ING. ENRIQUE REINOSA

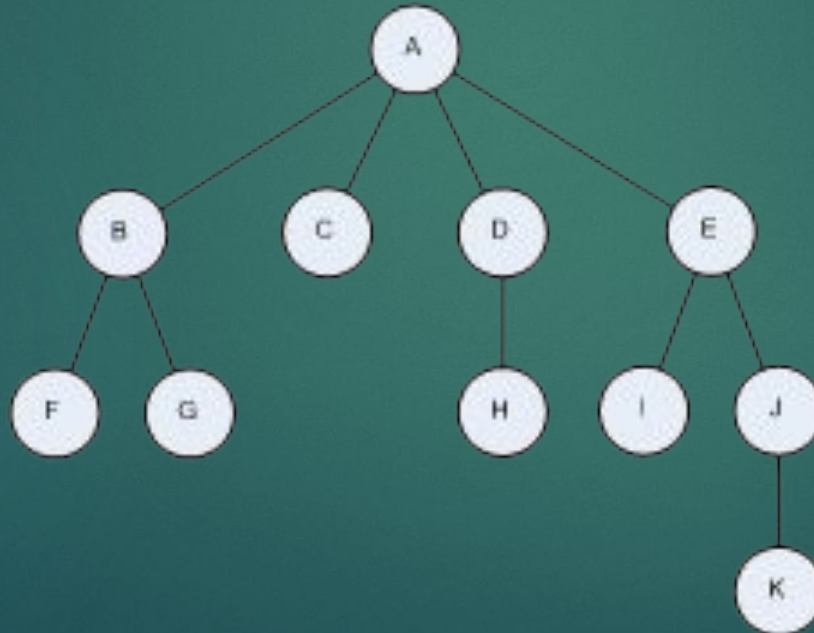
ARBOLES - CONCEPTOS

- ▶ **GRADO:** el grado de un árbol es por definición la **máxima cantidad de hijos o subárboles** que puede tener cada nodo.
- ▶ **NIVEL:** el nivel es la **posición en la que se encuentra cada nodo con respecto a la raíz** del mismo, considerando que la raíz se encuentra en el nivel 0



ARBOLES - PROFUNDIDAD

- ▶ **PRODUNDIDAD**: la profundidad de un árbol es la **cantidad de niveles** con que cuenta el mismo.

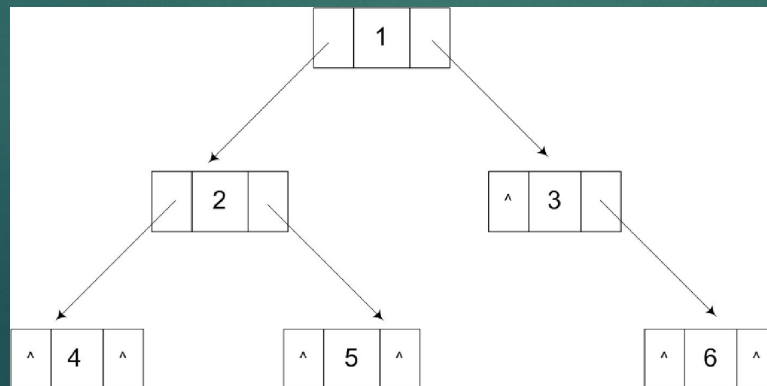


ARBOLES – REPRESENTACION COMPUTACIONAL

- ▶ **ESTATICA**: un árbol se representa en forma estática a través de un vector.



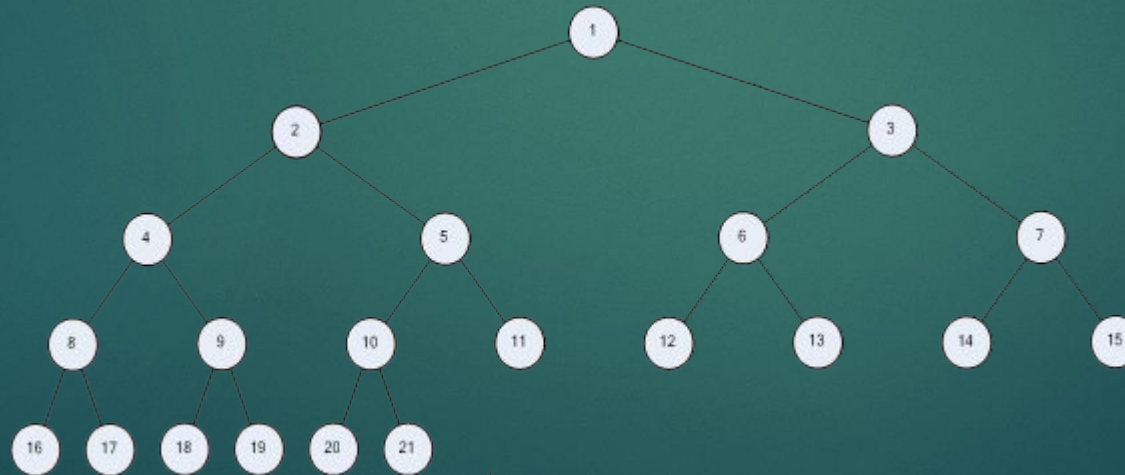
- ▶ **DINAMICA**: un árbol se representa en forma dinámica a través de un conjunto de nodos de igual tipo vinculados entre si a través de punteros o links



ARBOLES – CARACTERISTICAS

En los árboles se analizan características que no se analizan en otras estructuras de datos debido a que las restantes son biunívocas

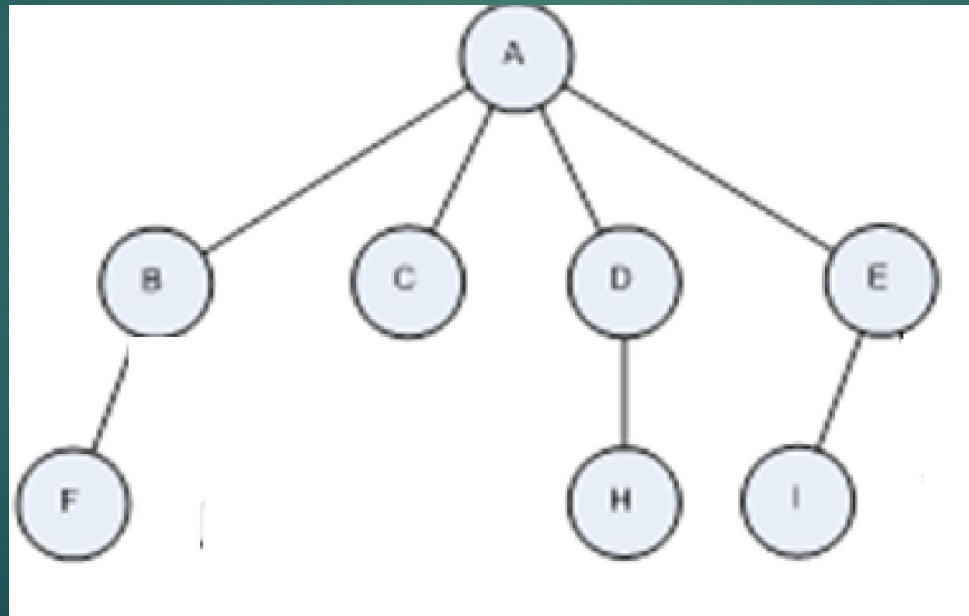
- ▶ **COMPLETO**: un árbol completo es aquel en el cual todos los nodos del árbol cumplen el grado o son hojas.



Árbol binario completo

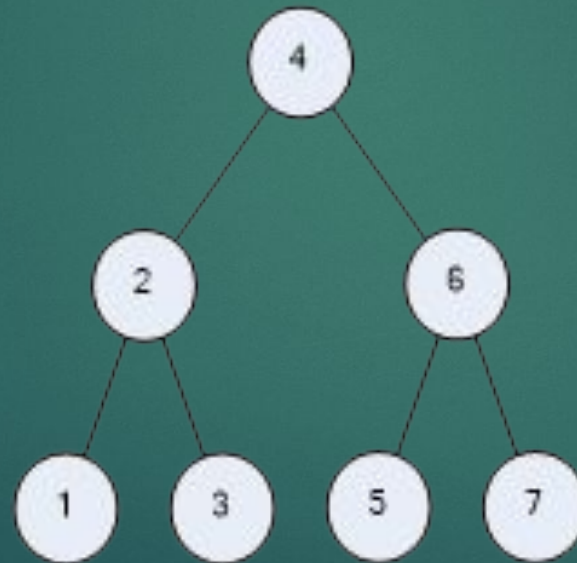
ARBOLES – CARACTERISTICAS

- ▶ **BALANCEADO**: un árbol está balanceado si **todos los subárboles desde la raíz pesan lo mismo** o sea tienen la misma cantidad de elementos **o una diferencia indivisible**.



ARBOLES – CARACTERISTICAS

- ▶ **PERFECTAMENTE BALANCEADO**: un árbol está perfectamente balanceado, cuando está **balanceado en todos sus niveles**-



Árbol binario perfecto

ARBOLES – CRECIMIENTO

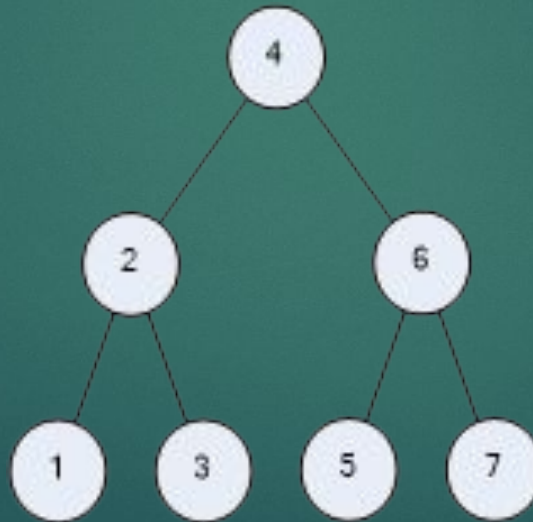
El crecimiento de un árbol es exponencial en función del grado del mismo, o sea, que en cada nivel puede crecer en función del grado definido.

La máxima cantidad de elementos posibles de un árbol esta da en función de la siguiente fórmula:

$$\text{Max Elementos} = \text{grado}^{\text{niveles} - 1}$$

ARBOLES – CRECIMIENTO

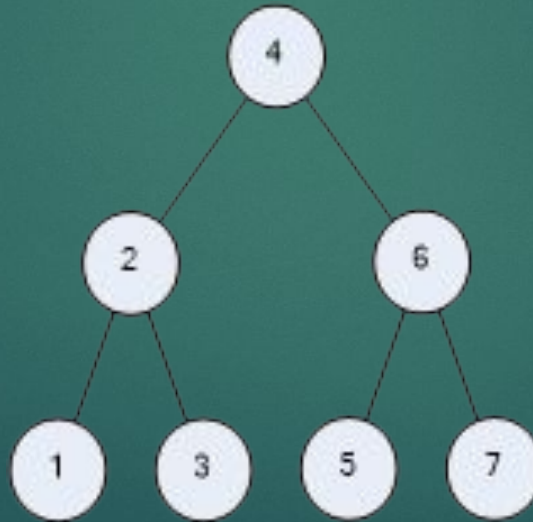
De esta forma si tomamos el árbol de la figura veremos que la máxima cantidad de elementos que puede tener es 7 que responde a $2^3 - 1$ que es el grado elevado a 3 que son los niveles menos 1 que está dado por el grado de inparidad de la raíz



Árbol binario perfecto

ARBOLES – BUSQUEDA

De la misma forma si se quisiera encontrar un elemento en dicho árbol, la búsqueda se realiza por niveles y no por elementos de forma tal que para encontrar cualquier elemento a lo sumo realizaremos tantas preguntas como niveles tenga el árbol. En este caso tres



Árbol binario perfecto

ARBOLES – BUSQUEDA

Si tomamos la fórmula de crecimiento y despejamos de ella la cantidad de niveles llegamos a la siguiente fórmula de búsqueda que justifica que **un árbol tiene una búsqueda logarítmica**

$$\text{elementos} = \text{grado}^{\text{niveles}} - 1$$

$$\text{elementos} + 1 = \text{grado}^{\text{niveles}}$$

$$\log \text{elementos} + 1 = \text{niveles} * \log \text{grado}$$

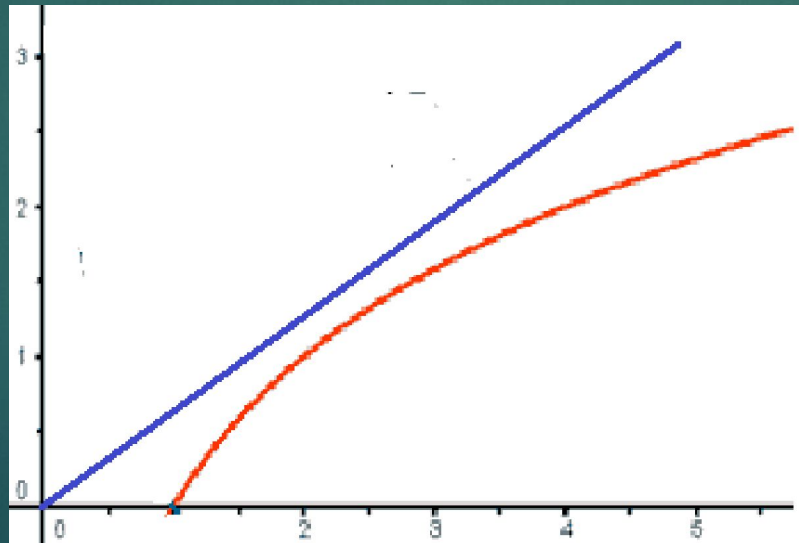
$$\text{niveles} = \log \text{elementos} + 1 - \log \text{grado}$$

$$\text{niveles} > \log \text{elementos}$$

ARBOLES – BUSQUEDA

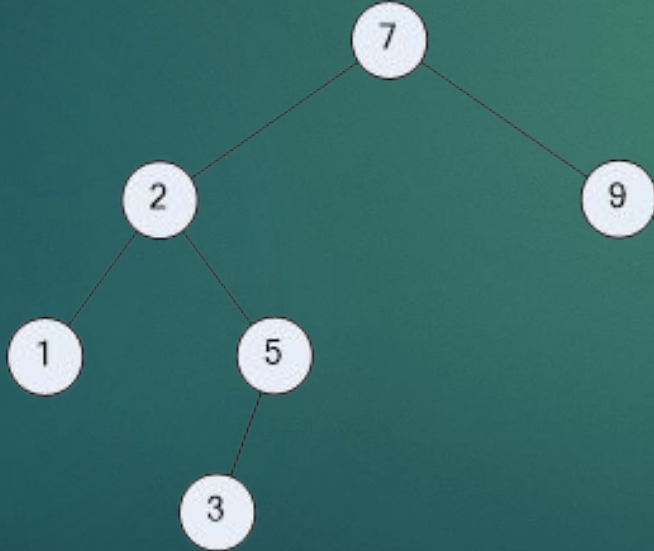
Una estructura de datos básica como por ejemplo una lista tiene una búsqueda secuencial, por lo tanto, si tengo 10 elementos a lo sumo debería realizar 10 preguntas para encontrar un elemento.

En un árbol esa búsqueda no es lineal, sino que es logarítmica

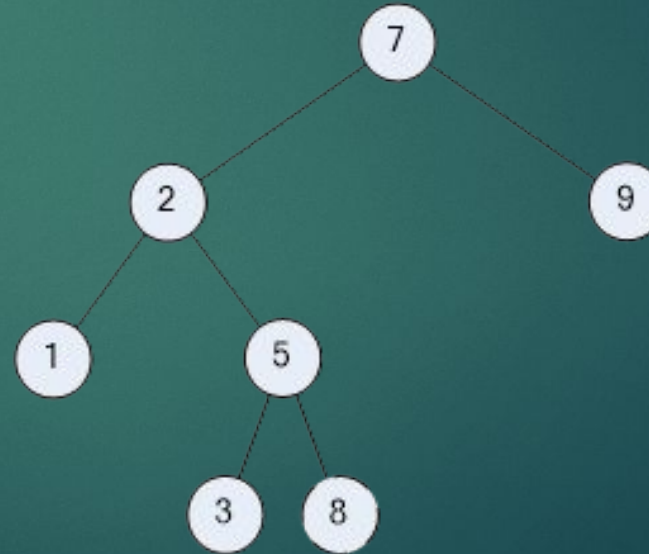


ARBOLES – ARBOL BINARIO DE BUSQUEDA (ABB)

- Un ABB como lo indica su nombre es un árbol de grado 2 diseñado para buscar como método alternativo a una lista representada en un vector, donde los elementos menores se ingresan a la izquierda y los mayores a la derecha



Árbol binario de búsqueda



Árbol binario

ARBOLES – ABB EJEMPLO

Dado el siguiente conjunto de números desordenados:

8 – 22 – 14 – 13 – 45 – 11 – 3 – 7 – 4 – 5 – 9

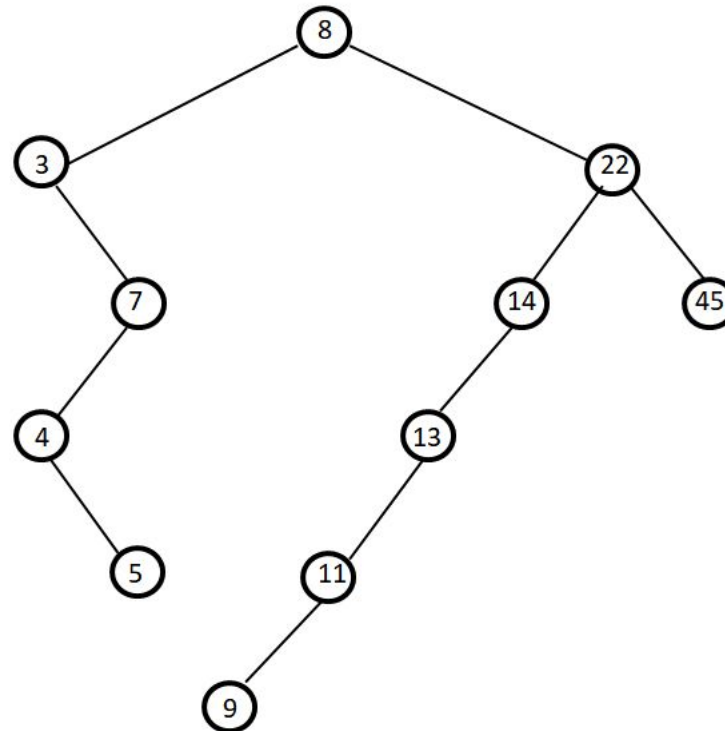
Si utilizamos una lista para ordenar los mismos, por ejemplo implementada en un vector nos quedaría lo siguiente:

3	4	5	7	8	9	11	13	14	22	45
---	---	---	---	---	---	----	----	----	----	----

ARBOLES – ABB EJEMPLO

Si utilizamos un ABB nos quedaría lo siguiente:

8 – 22 – 14 – 13 – 45 – 11 – 3 – 7 – 4 – 5 – 9



Mi interpretación es...

En la ppt va armando el arbol en base a como ya estan ordenados los numeros de la lista.

El 8 al ser el primero, será la raiz.

El 22 como es mayor a la raiz, lo pone en el subarbol derecho

El 14 como es mayor a la raiz, tambien lo pone en el subarbol der. pero como es menor que el 22, lo coloca en e subsubarbol izq.

Lo mismo con el 13

El 45 se repite, pero es mayor al 22 del subarbol der. entonces lo coloca en el subsubarbol derecho

El 3 es menor a la raiz (8) entonces lo coloca en el subarbol izq.

El 7 es menor a la raiz (8) entonces lo coloca en el subarbol izq. pero en el subsubarbol de derecho xq es mayor al 3

y asi.. arma el arbol por como estan ordenados los numeros en la lista, podria haber quedado de otra forma si no

ARBOLES – BARRIDOS

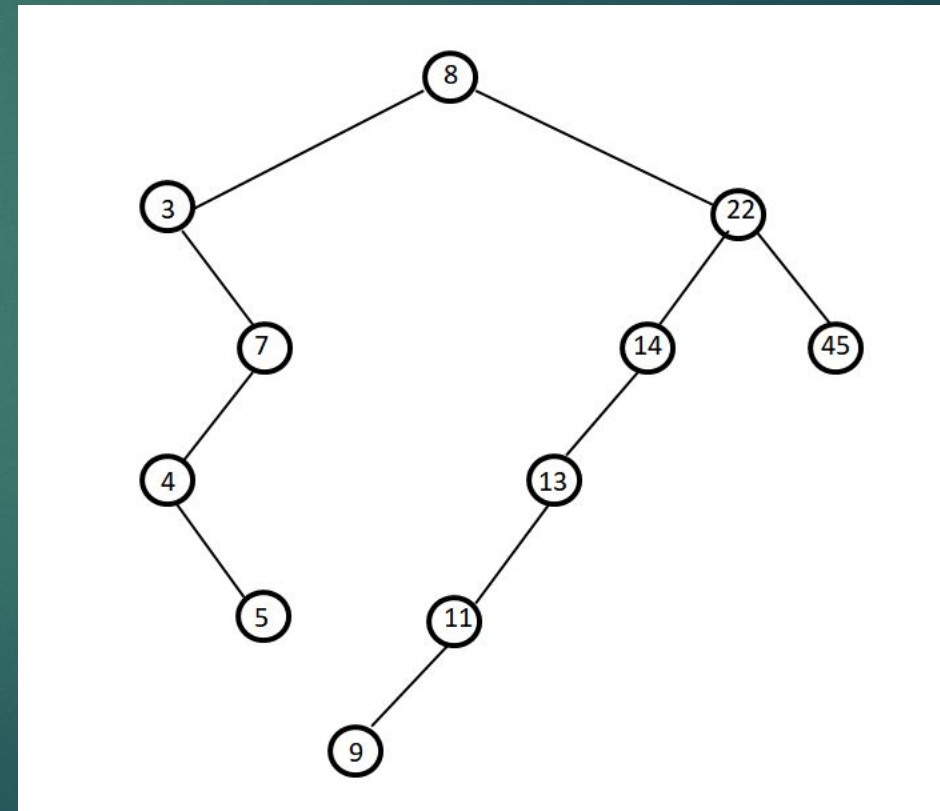
Un BARRIDO de un árbol **es la forma de leer el mismo**, si bien existe una forma de recorrer un árbol que es de arriba hacia abajo y de izquierda a derecha por convención occidental, **existen tres formas** distintas de leer los elementos que conforman tres barridos diferentes

- ▶ **Preorden**: el nodo se lee apenas se llega al mismo
- ▶ **Postorden**: el nodo se lee cuando se va del mismo y no se va regresar
- ▶ **Inorden**: el nodo se lee cuando se cambia de rama en el árbol

ARBOLES – BARRIDOS

Dado el siguiente árbol veremos cual es el resultado de cada barrido

- ▶ **Preorden:** 8-3-7-4-5-9-22-14-13-11-45
- ▶ **Postorden:** 9-5-4-7-3-11-13-14-45-22-8
- ▶ **Inorden:** 3-4-5-7-8-9-11-13-14-22-45



ARBOLES – BARRIDOS

El algoritmo de barridos es siempre igual, dado que un árbol se basa en la recursividad, lo único que se modifica es el momento en que se lee el nodo.

Preorden

if root

```
printf(root->dato);  
preorden(root->izq);  
preorden(root->der);  
return;
```

Inorden

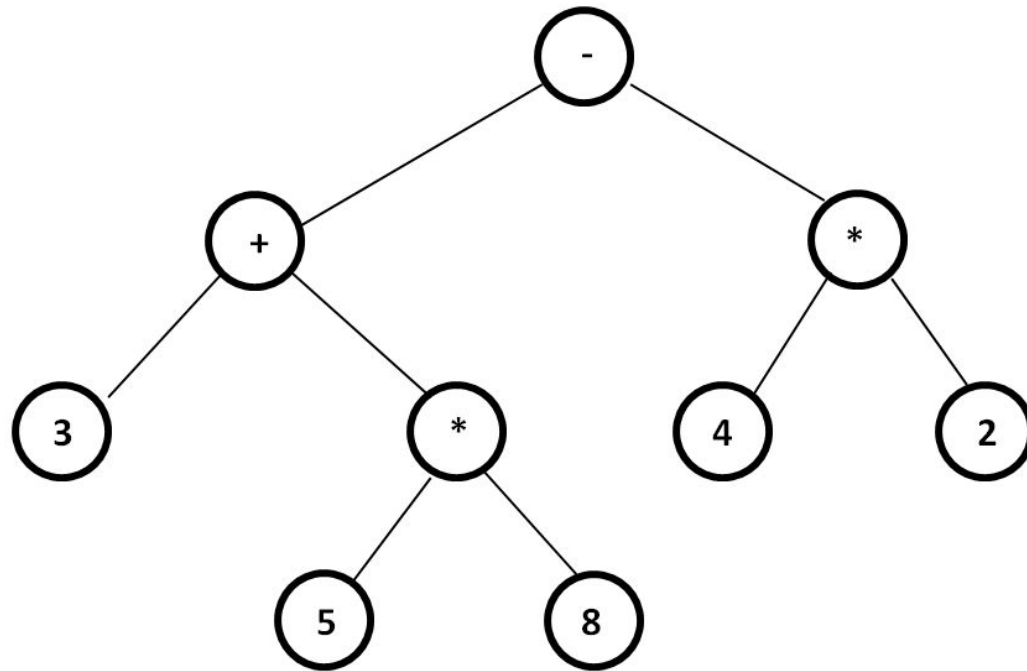
if root

```
inorden(root->izq);  
printf(root->dato);  
inorden(root->der);  
return;
```

ARBOLES – ARBOL DE EXPRESION

Una expresión puede representarse y resolverse a partir de un árbol, a estos se los llama árboles de expresión

$$3 + 5 * 8 - 4 * 2$$

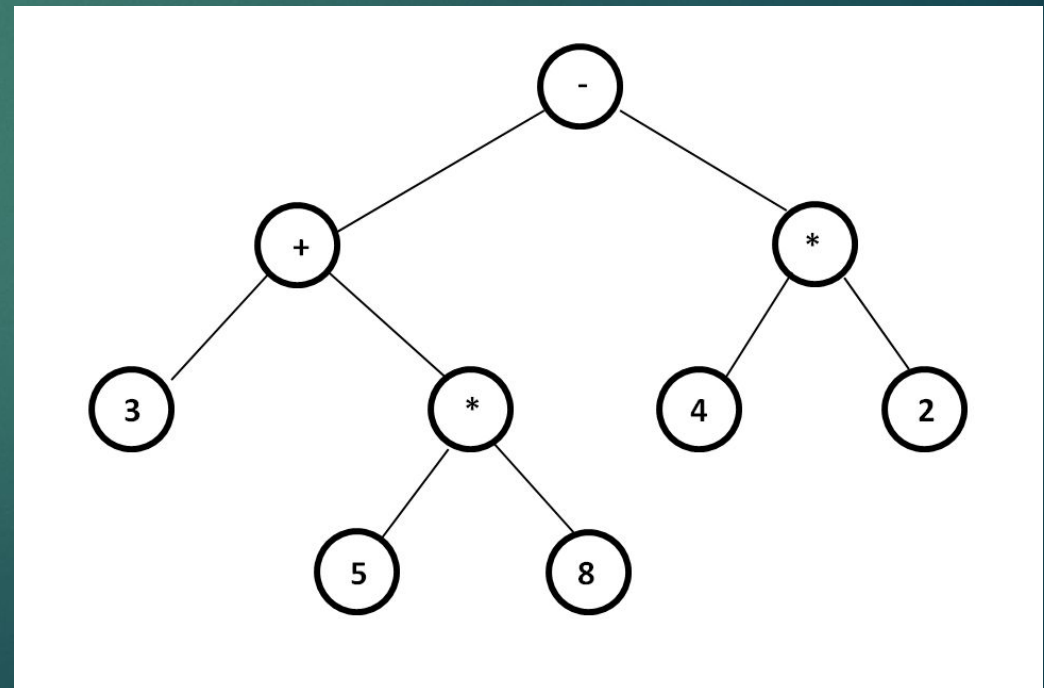


ARBOLES – ARBOL DE EXPRESION

Si a dicho árbol se barre inorder se obtiene la expresión matemática en notación INFIJO, si se lo barre postorden se obtiene la expresión matemática en notación POSTFIJO o POLACA INVERSA

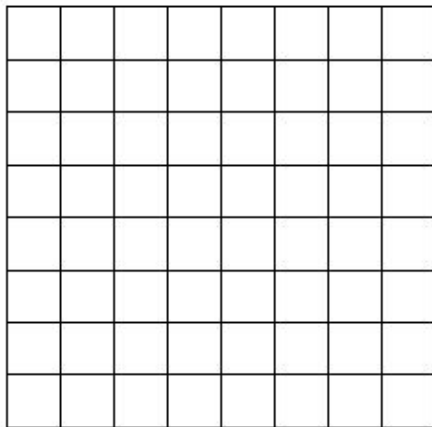
INORDEN: $3 + 5 * 8 - 4 * 2$

PORTORDEN: $358*+42*-$



SALTO DEL CABALLO

Existe un tablero de 8 x 8 conformado por 64 casilleros



Fila	Col
2	1
2	-1
-2	1
-2	-1
-1	2
-1	-2
1	2
1	-2

El caballo se mueve en “L” o sea en desplazamientos 2 x 1 o 1 x 2 en todos los sentidos, esto genera que existan 8 movimientos legales

El juego inicia de una posición que puede ser cualquier vértice o sea una posición (x,y) y desde ahí debe jugar a otra posición legal pero además posible, o sea, que caiga dentro del tablero y que no este ocupada.

El objetivo del juego es completar todo el tablero con números del 1 al 64 de forma tal que requiere moverse o jugar 64 veces.