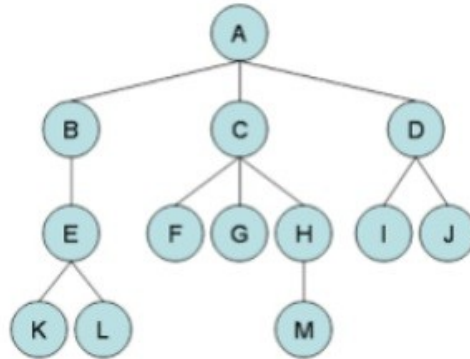


TP3 - Organización indexada – Arboles

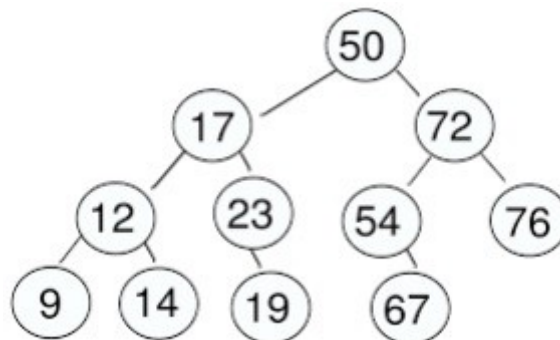
Alumno: Juan Cruz Mateos
Mat. 15134

1) Completar:



Raiz	A
Hermanos	(A), (B,C,D), (E), (F, G, H), (I, j), (K, L), (M)
Terminales	K, L, F, G, M, I, J
Interiores	B, E, C, H, D
Grado (nro descendientes directos nodo)	A:3, B:1, C:3, D:2, E:2, F:0, G:0, H:1,I:0, J:0, K:0, L:0, M:0
Grado del Arbol	3
Nivel (nro nodos que deben ser recorridos desde la raiz para llegar a un determinado nodo)	A:0, B:1, C:1, D:1, E:2, F:2, G:2, H:2,I:2, J:2, K:3, L:3, M:3
Altura (maximo numero de niveles de entre todas las ramas del arbol +1)	4

2) Completar y realizar recorridos : Preorden, Inorden, y Postorden



Preorden: 50, 17, 12, 9, 14, 23, 19, 72, 54, 67, 76
 Inorden: 9, 12, 14, 17, 23, 19, 50, 54, 67, 72, 76
 Postorden: 9, 14, 12, 19, 23, 17, 67, 54, 76, 72, 50

¿Es estricto? ¿Es lleno? ¿Es completo?

- *Estricto: No. Para ser estricto si un subarbol esta vacio, entonces el otro tambien debe estarlo. Es decir, un nodo puede tener 0 o 2 hijos unicamente. Pero los nodos 23 y 54 poseen solo un hijo a derecha, por lo tanto no es estricto.*
- *Lleno: No. Para el nodo 72 la altura del subarbol derecho e izquierdo no coinciden.*
- *Completo: No. Si bien es un arbol lleno hasta el penultimo nivel, en el ultimo nivel sus nodos no estan agrupados a la izquierda.*

3) Diseñe la estructura para implementar un árbol binario que contenga como datos números enteros. Implemente funciones para agregar nodos, recorrer inorden y buscar un elemento en particular (en forma recursiva e iterativa).

```

typedef struct nodo {
    int clave;
    struct nodo *izq;
    struct nodo *der;
} nodo_t;

typedef nodo_t *arbol_t;

```

4) Realice las rutinas necesarias para agregar 100.000 elementos al azar en el árbol.

5) Busque un elemento cualquiera (conceptualmente) ¿Cuántas comparaciones fueron necesarias en promedio?

6) Plantee (implemente) una solución de persistencia para su árbol. ¿Con qué problemas se encontró?

7) Realice los puntos 3, 4, 5 y 6 pero utilizando montículos.

ÁRBOLES B, B* y B+

8) Dada la secuencia de claves enteras: 190, 57, 89, 90, 121, 170, 35, 48, 91, 22, 126, 132 y 80; dibuje el árbol B de orden 2 ($2d+1$ punteros) cuya raíz es R, que se corresponde con dichas claves.

9) En el árbol R del problema anterior, elimine la clave 91 y dibuje el árbol resultante. Elimine ahora la clave 48. Dibuje el árbol resultante, ¿se redujo el número de nodos?

10) Dada la siguiente secuencia de claves: 7, 25, 27, 15, 23, 19, 14, 29, 10, 50, 18, 22, 46, 17, 70, 33 y 58 dibuje el árbol B+ cuya raíz es R, que se corresponde con dichas claves.

11) Construir cada uno de los arboles B (B, B+ y B*) que se van generando conforme se van insertando los números 1, 9, 32, 3, 53, 43, 44, 57, 67, 7, 45, 34, 12, 23, 56, 73, 65, 49, 85, 89, 64, 54, 75, 77 en un árbol B de orden 5.