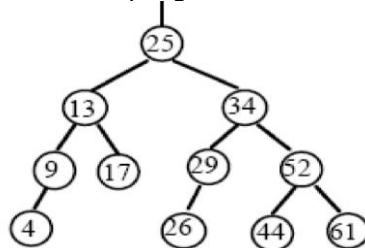


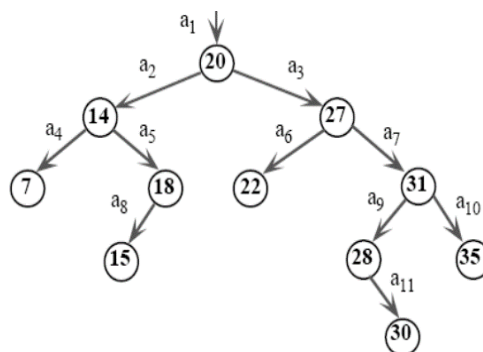
Árboles Binarios de Búsqueda BST

- 1) Ingrese el siguiente BST en su el programa realizado en clase:



Programa las siguientes funciones:

- Recorrido preorder, en este tipo de recorrido se recorre primero la raíz, luego la rama izquierda y para finalizar la rama derecha.
 - Recorrido postorder, en este recorrido se recorre primero la parte izquierda, luego la parte derecha y al final la raíz.
- 2) Considere el BST de la pregunta 1, realice una función que permita encontrar un elemento n al interior de dicho árbol.
- 3) Formule un algoritmo eficiente que determine el elemento máximo y el elemento mínimo de en un BST. Programe su solución.
- 4) Dado el siguiente ABB:
- Muestre los pasos a seguir para eliminar el nodo 30.
 - Muestre los pasos a seguir para eliminar el nodo 18.
 - Muestre los pasos a seguir para eliminar el nodo 27.



Haga una prueba de escritorio, utilizando el programa desarrollado en clases, que permita resolver las preguntas a) b) y c).

- 5) Considere la información relacionada a los productos de una gran cadena comercial almacenados en un árbol binario de búsqueda; los campos a considerar son los siguientes:

Código del Producto	(CP)
Nombre del Producto	(NP)
Cantidad en Stock	(STOCK)

Asuma que la clave de ordenamiento es el campo Código del Producto. Formule un algoritmo recursivo y un programa para cada una de las siguientes operaciones:

- a) La operación **MostrarListado**, que consiste en mostrar un listado ordenado por código de producto con su correspondiente nombre de producto y cantidad en stock.
 - b) La operación **ActualizarStock**, que consiste en actualizar el stock en M unidades adicionales al producto cuyo código es COD.
 - c) La operación **ContarProductos**, que consiste en contar todos los productos que tienen una cantidad en stock superior a CANT.
- 6) Considere la información relacionada a la cantidad de pacientes asignados para ser atendidos a los médicos de un hospital. Dicha información se encuentra almacenada en un BST. Los campos a considerar son los siguientes:
- Código del Doctor (CD)
 Especialidad (ESP)
 Cantidad de Pacientes (CPAC)

Asuma que la clave de ordenamiento es el campo Código del Doctor. Formule un algoritmo y un programa para cada una de las siguientes operaciones:

- a) La operación **EscribirListado**, que consiste en mostrar el código de doctor y la cantidad de pacientes que tiene, ordenado por código de doctor, correspondiente a todos los doctores de la especialidad EE. Formule un algoritmo recursivo.
 - b) La operación **ActualizarCantPacientes**, que consiste en añadir la cantidad PP de pacientes al doctor cuyo código es DOC. Formule un algoritmo recursivo.
 - c) La operación **ContarPacientes**, que consiste en contar todos los pacientes que son atendidos por médicos cuya especialidad sea EE. Formule un algoritmo no recursivo.
- 7) Considere la información relacionada a las ventas anuales de libros publicados por profesores de la Universidad de Lima; dicha información se encuentra almacenada en un BST. La estructura de cada nodo del árbol es la siguiente:

Nombre del Docente (ND)
 Título del Libro (TL)
 Cantidad vendida (CV)
 Precio Unitario (S/.) (PU)

Asuma que la clave de ordenamiento es el campo Nombre del Docente. Formule algoritmos recursivos y un programa que realicen las siguientes tareas:

- a) Mostrar un listado ordenado por nombre de docente con la correspondiente venta total de sus libros (cantidad vendida x precio unitario).
 - b) Mostrar un listado de títulos de libros y cantidades vendidas correspondientes al docente DD.
 - c) Mostrar un listado de todos los docentes que tienen una venta superior a VV.
- 8) Escriba un algoritmo no recursivo que permita calcular el nivel en que se encuentra un valor x en un árbol binario de búsqueda. Suponga que x se encuentra en el árbol.
- 9) Escriba un programa recursivo que permita obtener la altura de un árbol binario. La altura de un árbol se define como la cantidad de niveles de un árbol. Un algoritmo para resolver este problema es el siguiente:

```
altura(raiz)
if raiz=nulo
  retornar -1
else
  retornar 1+maximoValor(altura(raiz.izq),altura(raiz.der))
endif
```

En el caso de la función `maximoValor` puede usar la librería `Math.max` de Java. Adicionalmente haga una prueba de escritorio (<https://aztlek.org/2014/08/11/como-hacer-una-prueba-de-escritorio/>) e indique porqué en el caso base cuando la raíz es nula se retorna un -1.

- 10) Realice un programa iterativo que devuelva la altura de un árbol binario. Un probable algoritmo sería el siguiente:

```
alturaIter(raiz)
  if raiz=null return 0
  //q es una cola
  q.encolar(raiz)
  alt=-1
  while true
    nodoC=q.size
    if nodoC=0 return alt
    alt=alt+1
    while nodoC>0
      nodo=q.decolar()
      if nodo.izq!=null
        q.encolar(nodo.izq)
      endif
      if nodo.der!=null
        q.encolar(nodo.der)
      endif
      nodoC=nodoC-1
    endwhile
  endwhile
```

Haga la prueba de escritorio para un BST cualquiera.

- 11) Realice un programa que permita eliminar el elemento mínimo de un árbol.