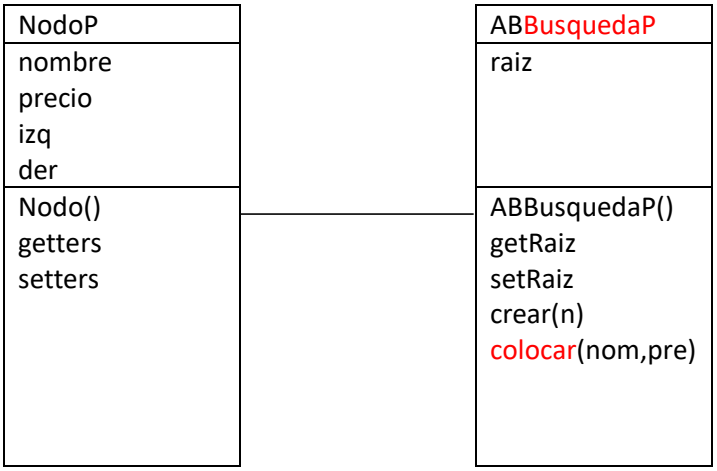
Sea un arbol Producto(nombre, precio), organizado por el precio




```
Class NodoP
{  private
       String nombre
       Int precio
       NodoP izq,der
   Public
      NodoP()
      {izq=der=null}
       Getters
        setters
}
```

```
Class ABBusquedaP
{  private
      NodoP raíz
  Public
      ABBusquedaP()
      {
          Raíz=null
      }
      getRaiz/setRaiz
      colocar(string nom,int pre)
      {
         nue=new NodoP()
         nue.setNombre(nom)
         nue.setPrecio(pre)
         If getRaiz()=null
              setRaiz(nue)
         else{
            NodoP w=getRaiz()
            Sw=true
        While ,sw
        {    If pre>w.getPrecio()
             {   if w.getDer()<>null
                      w=w.getDer()
                 else
                      { w.setDer(nue); sw=false}
             }
              Else
                If pre<w.getPrecio()
                {  if w.getIzq()<>null
                        w=w.getIzq()
                   else
                        {w.setIzq(nue);sw=false}
                }
                else /// es igual
                   { Print("ya existe..");sw=false}
           }//while
         }
      }
   Crear(int n)
   {
       For i=1 to n
       {  read(no,pr)
          colocar(no,pr)
```

```
        }
    }
    buscar(int x)
    {  NodoP  z=getRaiz()
       Sw=false
       While  z<>null{
          If x>z.getPrecio()
              z=z.getDer()
          else
              if x<z.getPrecio()
                 z=z.getIzq()
              else//igual
                 {   print(z.getNombre()); sw=true;z=null}
       }//while
       If sw
           Print( "existe")
       Else
          Print("no existe")
    }
  mayor()
  {  NodoP w=getRaiz()
     while w.getDer<>null
         w=w.getDer()
     print(w.getNombre())
  }
  Menor()
  {  NodoP w=getRaiz()
     While w.getIzq()<>null
           w=w.getIzq()
     print(w.getNombre())
  }
}
```

Sea el árbol **A** que almacena en cada nodo un número.

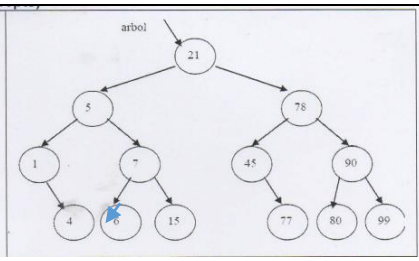
Mostrar el mayor y menor del dato X, si existe.

Ejemplos:

Si X=78 mayor=99 menor=45

Si X=45 mayor=77 menor=45

Si X=6 mayor=6 menor=6

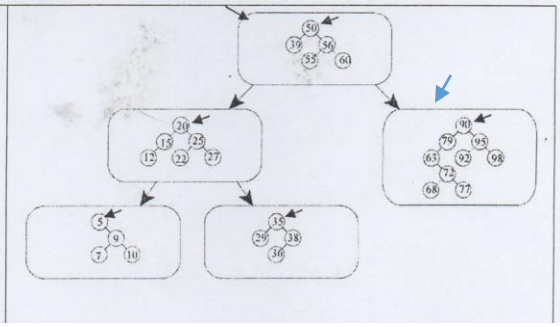



```
Solucion(int x)
{   NodoP w=getRaiz()
    While w<>null
    {
        If x>w.getPrecio()
            w=w.getDer()
        else
            if x<w.getPrecio()
                w=w.getIzq()
            else //igual
            {  print(may(w),men(w))
               w=null
            }
    }
}
may(NodoP r)
{  while r.getDer()<>null
       r=r.getDer()
   return (r.getPrecio())
}
Men(NodoP r)
{  while r.getIzq()<>null
       r=r.getIzq()
   return (r.getPrecio())
 }
```

a. Establecer el diagrama de clases
b. Verificar si existe el dato X.
c. Verificar si el dato X está en la raíz de algún árbol

Se asume la clase implementada ArbolBB

ArbolBB
ArbolBB() Adicionar(x) Eliminar(x) Verificar(x) Maximo(x) Minimo(x) Buscar(x)



```
Class bosque
{   private
        Nodo raiz
    Public
      VerificaB(int x)
      {
         Nodo r=getRaiz()
         While r<>null
         {  if x>r.getA().Maximo()
                r=r.getDer()
            else
              if x<r.getA().Minimo()
                  r=r.getIzq()
              else{
                    if r.getA().Verificar(x)
                        print("existe")
                    else
                        print("no existe")
               r=null
               }
         }
      }

}
```

```
Class Nodo
{ private
        ArbolBB a
        Nodo izq,der
   Public

}
```