UNIVERSIDAD MAYOR DE SAN "ANDRÉS"
CARRERA DE INFORMÁTICA
PROGRAMACIÓN II
ÁRBOLES

Responsable: Lic. Victoria Hurtado Cerruto Gestion:2/2024

10 PROBLEMAS RESUELTOS

PROBLEMA 1

Dado un ABNormal se pide eliminar aquellos nodos terminales donde su nodo hermano no es terminal.

DESARROLLO LÓGICO

Dado un árbol binario normal ***x*** extraemos su raíz en la variable ***rx*** y a partir de ella empezamos analizar todos los nodos del árbol. Adicionalmente utilizaremos una Pila ***aux*** donde almacenaremos los lados derechos de cada nodo procesado.

En sí, para nodo***rx***del árbol verificamos si tiene tanto el descendiente izquierdo (**si)** y descendiente derecho (**sd)** en tal caso accedemos a dichos descendientes e identificamos dos casos:

- El caso de que el descendiente izquierdo ***si***sea hoja y el descendiente derecho***sd***no sea hoja en tal caso se desconecta el lado izquierdo del nodo analizado.

- El caso de que el descendiente derecho ***sd*** sea hoja y el descendiente izquierdo ***si*** no sea hoja en tal caso se desconecta el lado derecho del nodo analizado.

CODIFICACIÓN

```
public class EliTerconHernoTer {
public static void main(String[] args) {
ABNormal x=new ABNormal();
NodoAr rx,si,sd;
x.llenar();
x.m_Nivel();
rx=x.getRaiz();
Pila aux=new Pila(100);
aux.adicionar(null);
while(rx!=null)
{
 if((rx.getSi()!=null)&&(rx.getSd()!=null))
 {
  si=rx.getSi();
  sd=rx.getSd();
  if
(((si.getSi()==null)&&(si.getSd()==null))&&((sd.getSi()!=null)||(sd.ge
tSd()!=null))) rx.setSi(null);
            else
            if
(((sd.getSi()==null)&&(sd.getSd()==null))&&((si.getSi()!=null)||(si.ge
tSd()!=null))) rx.setSd(null);
         }
if (rx.getSd()!=null)aux.adicionar(rx.getSd());
if (rx.getSi()!=null)rx=rx.getSi();
else rx=(NodoAr)aux.eliminar();
    }
```

```
        x.m_Nivel();
}
}
```

TRAZA DE LA EJECUCIÓN

```
Digite dato de la raiz
5
Desea descendiente izquierdo de 5 Si=1 No<>1?1
Digite dato izquierdo
7
Desea descendiente derecho de 5 Si=1 No<>1?1
Digite dato derecho
4
Desea descendiente izquierdo de 7 Si=1 No<>1?1
Digite dato izquierdo
2
Desea descendiente derecho de 7 Si=1 No<>1?1
Digite dato derecho
9
Desea descendiente izquierdo de 4 Si=1 No<>1?1
Digite dato izquierdo
3
Desea descendiente derecho de 4 Si=1 No<>1?1
Digite dato derecho
8
Desea descendiente izquierdo de 2 Si=1 No<>1?2
Desea descendiente derecho de 2 Si=1 No<>1?2
Desea descendiente izquierdo de 9 Si=1 No<>1?1
Digite dato izquierdo
10
Desea descendiente derecho de 9 Si=1 No<>1?2
Desea descendiente izquierdo de 3 Si=1 No<>1?1
Digite dato izquierdo
6
Desea descendiente derecho de 3 Si=1 No<>1?2
Desea descendiente izquierdo de 8 Si=1 No<>1?2
Desea descendiente derecho de 8 Si=1 No<>1?2
Desea descendiente izquierdo de 10 Si=1 No<>1?2
Desea descendiente derecho de 10 Si=1 No<>1?2
Desea descendiente izquierdo de 6 Si=1 No<>1?2
Desea descendiente derecho de 6 Si=1 No<>1?2
5
7       4
2       9       3       8
10      6
5
7       4
9       3
10      6
```

PROBLEMA 2

Dado un árbol binario de búsqueda se pide eliminar los nodos terminales.

DESARROLLO LÓGICO

Para cada nodo del árbol se verifica si tiene descendientes en tal caso se verifica si estos descendientes son hojas en tal caso se desconecta del nodo respectivo ya sea del descendiente izquierdo o bien del descendiente derecho.

UNIVERSIDAD MAYOR DE SAN "ANDRÉS"
CARRERA DE INFORMÁTICA
PROGRAMACIÓN II
ÁRBOLES

Responsable: Lic. Victoria Hurtado Cerruto Gestion:2/2024

CODIFICACIÓN

```
public class BorraTerminal {
public static void main(String[] args) {
ABB y=new ABB();
  NodoAr rx,si,sd;
  y.llenar();
y.m_Nivel();
  rx=y.getRaiz();
  Pila aux=new Pila(100);
  aux.adicionar(null);
while(rx!=null)
  {
if((rx.getSi()!=null)&&(rx.getSd()!=null))
   {
    si=rx.getSi();
    sd=rx.getSd();
if ((si.getSi()==null)&&(si.getSd()==null)) rx.setSi(null);
else
if ((sd.getSi()==null)&&(sd.getSd()==null))rx.setSd(null);
    }
if (rx.getSd()!=null)aux.adicionar(rx.getSd());
if (rx.getSi()!=null)rx=rx.getSi();
else rx=(NodoAr)aux.eliminar();
  }
y.m_Nivel();
 }
}
```

TRAZA DE LA EJECUCIÓN

```
Número de elementos?
6
Ingrese dato entero al árbol?
20
Ingrese dato entero al árbol?
8
Ingrese dato entero al árbol?
40
Ingrese dato entero al árbol?
15
Ingrese dato entero al árbol?
70
Ingrese dato entero al árbol?
2
20
8       40
2       15      70
20
8       40
15      70
```