

Estructuras de Datos Orientadas a Objetos

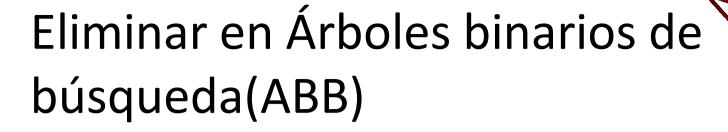
Profesor: Aldonso Becerra Sánchez



Árboles binarios de búsqueda(ABB)

Para todo nodo T del árbol se debe cumplir que todos los valores almacenados en el subárbol izquierdo de T sean menores o iguales a la información guardada en el nodo T. De forma similar, todos los valores almacenados en el subárbol derecho deben ser mayores o iguales a la información guardada en el nodo T.

Propósito: buscar, insertar y eliminar de manera eficiente.



- 1. Si el elemento es una hoja o terminal, simplemente se suprime eliminando el puntero de su predecesor.
- 2. Si el elemento a borrar tiene un solo descendiente, entonces se sustituye por ese descendiente.
- 3. Si el elemento a borrar tiene los dos descendientes, entonces se tiene que sustituir por el nodo que se encuentra más a la izquierda del subárbol derecho o por el nodo que se encuentra más a la derecha del subárbol izquierdo.

Eliminación en ABB



eliminacionABB(apnodo,anterior,info)

```
1. Si (apnodo<>nil) entonces
   1.1 Si(info<apnodo^.info) entonces
             Llamar a eliminacionABB con apnodo^.izg,apnodo e info
       si no {del 1.1}
       1.1.1 Si(info>apnodo^.info) entonces
                Llamar a eliminacionABB con apnodo.der, apnodo e info
           si no {del 1.1.1, es igual al nodo a eliminar }
                1.1.1.1. Si (apnodo^.izq<>nil and apnodo^.der<>nil)
                  {tiene dos hijos, saca el más a la derecha del subarbol izquierdo}
                    aux←apnodo^.izq y bo ← falso
                    1.1.1.1. mientras(aux^.der<>nil)
                              aux1←aux, aux←aux^.der y bo←verdadero
                    1.1.1.1.2. {fin de paso 1.1.1.1.1}
                    apnodo^.info← aux^.info
                    1.1.1.3. Si(bo=verdadero)
                              aux1^.der← aux^.izg
                    sino {del 1.1.1.1.3}
                              apnodo^.izq←aux^.izq
                    1.1.1.1.4. {fin del paso 1.1.1.1.3}
```

Eliminación en ABB



eliminacionABB(APNODO,ANTERIOR, INFO)... continúa

```
sino {del 1.1.1.1, el derecho, izquierdo o ambos son nulos }
          otro←apnodo
          1.1.1.5. Si(otro^.der=nil)
            1.1.1.5.A Si(otro^.izq<>nil){ sólo liga derecha es nulo}
                      otro←apnodo^.izq
                      1.1.1.5.A.1 Si (anterior<>nil)
                         1.1.1.1.5.A.1.1 Si(info<anterior^.info)
                                 anterior^.izg←otro
                         sino {del 1.1.1.5.A.1.1}
                                 anterior^.der←otro
                         1.1.1.1.5.A.1.2 {fin de 1.1.1.1.5.A.1.1}
                      sino {del 1.1.1.5.A.1}
                        raiz←otro
                      1.1.1.1.5.A.2 {fin de 1.1.1.1.5.A.1}
```

Eliminación en ABB eliminacion ABB (APNODO, ANTERIOR, INFO)... continua



```
sino{ del 1.1.1.1.5.A, ambos son nulos}
                1.1.1.1.5.A.4. Si(anterior=nil) {raíz solamente} raiz←nil
                sino {del 1.1.1.1.5.A.4}
                           1.1.1.1.5.A.4.1 Si(info<anterior^.info)
                             anterior^.izg← nil
                           sino {del 1.1.1.1.5.A.4.1}
                             anterior^.der←nil
                          1.1.1.1.5.A.4.2 {fin de 1.1.1.1.5.A.4.1}
      1.1.1.5.B. {fin de 1.1.1.1.5.A}
sino { del 1.1.1.1.5, la derecha no es nula}
      1.1.1.1.5.C. Si(otro^.izq=nil){ sólo liga izquierda es nulo} otro←apnodo^.der
                1.1.1.1.5.C.1 Si (anterior<>nil)
                  1.1.1.1.5.C.1.1 Si(info<anterior^.info) anterior^.izg←otro
                    sino {del 1.1.1.1.5.C.1.1} anterior^.der←otro
                  1.1.1.1.5.C.1.2 (fin de 1.1.1.1.5.C.1.1)
                sino {del 1.1.1.1.5.C.1}
                   raiz←otro
                1.1.1.1.5.C.2 (fin de 1.1.1.1.5.C.1)
```

Eliminación en ABB



eliminacionABB(APNODO,ANTERIOR, INFO)...continua

```
1.1.1.1.6 {fin de 1.1.1.1.5}
1.1.1.2 {fin de 1.1.1.1 del que quiere decir que no tiene los dos hijos}
1.1.2 }{fin de 1.1.1 de si es mayor, es decir que es igual}
1.2 {fin de 1.1 del de es menor}
sino { del 1}
Escribir "El dato no está en el árbol"
2. {fin de paso 1}
```