

```

void eliminarDerecha(ABB &arbol, int x){
    pNodo padre = NULL;
    pNodo actual = arbol;
    pNodo nodo;
    int aux;

    while(actual != NULL){
        if(x == actual->idNodo){
            if(actual->der == NULL && actual->izq == NULL){
                if(padre != NULL){
                    if(padre->der == actual){
                        padre->der = NULL;
                    }else{
                        if(padre->izq == actual){
                            padre->izq = NULL;
                        }
                    }
                }
                free(actual);
                actual = NULL;
            }else{
                padre = actual;
                if(actual->der != NULL){
                    nodo = actual->der;
                    while(nodo->izq != NULL){
                        padre = nodo;
                        nodo = nodo->izq;
                    }
                }else{
                    nodo = actual->izq;
                    while(nodo->der != NULL){
                        padre = nodo;
                        nodo = nodo->der;
                    }
                }
                aux = actual->idNodo;
                actual->idNodo = nodo->idNodo;
                nodo->idNodo = aux;
                actual = nodo;
            }
        }else{
            padre = actual;
            if(x > actual->idNodo){
                actual = actual->der;
            }else{
                if(x < actual->idNodo){
                    actual = actual->izq;
                }
            }
        }
    }
}

```

Eliminar por la Derecha

Nodo 3: 27  
 Nodo 4: 89  
 Nodo 5: 58  
 Nodo 6: 91  
 Nodo 7: 100  
 Nodo 8: 39  
 Nodo 9: 30  
 Nodo 10: 54  
 Nodo 11: 16  
 Nodo 12: 23  
 Nodo 13: 66  
 Nodo 14: 10  
 Nodo 15: 12

Algoritmos de Recorrido PreOrden:

53	27	16	10	12	23	39	30	61	58	54	89	66	91	100
----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Algoritmo de Recorrido InOrden:

10	12	16	23	27	30	39	53	54	58	61	66	89	91	100
----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Algoritmo de Recorrido PosOrden:

12	10	23	16	30	39	27	54	58	66	100	91	89	61	53
----	----	----	----	----	----	----	----	----	----	-----	----	----	----	----

Algoritmo de Eliminar por la derecha:

Algoritmos de Recorrido PreOrden:

54	27	16	10	12	23	39	30	61	58	89	66	91	100
----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Eliminar por la  
Izquierda

```
void eliminarIzquierda(ABB &arbol, int x){
    pNodo padre = NULL;
    pNodo actual = arbol;
    pNodo nodo;
    int aux;

    while(actual != NULL){
        if(x == actual->idNodo){
            if(actual->der == NULL && actual->izq == NULL){
                if(padre != NULL){
                    if(padre->der == actual){
                        padre->der = NULL;
                    }else{
                        if(padre->izq == actual){
                            padre->izq = NULL;
                        }
                    }
                }
                free(actual);
                actual = NULL;
            }else{
                padre = actual;
                if(actual->izq != NULL){
                    nodo = actual->izq;
                    while(nodo->der != NULL){
                        padre = nodo;
                        nodo = nodo->der;
                    }
                }else{
                    nodo = actual->der;
                    while(nodo->izq != NULL){
                        padre = nodo;
                        nodo = nodo->izq;
                    }
                }
                aux = actual->idNodo;
                actual->idNodo = nodo->idNodo;
                nodo->idNodo = aux;
                actual = nodo;
            }
        }else{
            padre = actual;
            if(x > actual->idNodo){
                actual = actual->der;
            }else{
                if(x < actual->idNodo){
                    actual = actual->izq;
                }
            }
        }
    }
}
```

Nodo 3: 27  
Nodo 4: 89  
Nodo 5: 58  
Nodo 6: 91  
Nodo 7: 100  
Nodo 8: 39  
Nodo 9: 30  
Nodo 10: 54  
Nodo 11: 16  
Nodo 12: 23  
Nodo 13: 66  
Nodo 14: 10  
Nodo 15: 12

Algoritmos de Recorrido PreOrden:

53	27	16	10	12	23	39	30	61	58	54	89	66	91	100
----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Algoritmo de Recorrido InOrden:

10	12	16	23	27	30	39	53	54	58	61	66	89	91	100
----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Algoritmo de Recorrido PosOrden:

12	10	23	16	30	39	27	54	58	66	100	91	89	61	53
----	----	----	----	----	----	----	----	----	----	-----	----	----	----	----

Algoritmo de Eliminar por la izquierda:

Algoritmos de Recorrido PreOrden:

53	27	16	10	12	23	39	30	61	58	54	89	66	100
----	----	----	----	----	----	----	----	----	----	----	----	----	-----