

Tareas

- Construir un árbol que permita ordenar una lista de datos
- Buscar un dato en el árbol ordenado para ver si se encuentra.
- Recorrer un árbol en orden – preorden – posorden
- Revisar si un árbol es estrictamente binario.
- Construir una función que diga si 2 árboles binarios son idénticos.
- Construir una función que encuentre el mayor dato dentro de un árbol ordenado.

Construcción de Arbol Ordenado

```
Func ArbolOrdenado( arbol, dato )
```

```
Si arbol == null
```

```
    arbol = MakeTree( dato )
```

```
Sino
```

```
    Si dato > info( arbol )
```

```
        Si isRight( arbol )
```

```
            ArbolOrdenado( right( arbol ), dato )
```

```
        Sino
```

```
            SetRight( arbol, dato )
```

```
    Sino
```

```
        Si isLeft( arbol )
```

```
            ArbolOrdenado( left( arbol ), dato )
```

```
        Sino
```

```
            SetLeft( arbol, dato )
```

Buscar un dato en el árbol

```
Func BuscarDato( arbol, dato )
```

```
Si info( arbol ) = dato
```

```
    Print "el dato se encuentra en el arbol"
```

```
Sino
```

```
    Si info( arbol ) > dato
```

```
        Si isLeft( arbol )
```

```
            BuscarDato( left( arbol ), dato )
```

```
        Sino
```

```
            Print " el dato NO se encuentra en el arbol"
```

```
Sino
```

```
    Si isRight( arbol )
```

```
        BuscarDato( right( arbol ), dato )
```

```
    Sino
```

```
        Print " el dato NO se encuentra en el arbol"
```

Recorrido de Arbol

```
Func RecArbolOrden( arbol )
```

```
Si IsLeft( arbol ) RecArbolOrden( left( arbol ) )
```

```
Print( info ( arbol ) )
```

```
Si IsRight( arbol ) RecArbolOrden( right( arbol ) )
```

```
Func RecArbolPreOrden( arbol )
```

```
Print( info ( arbol ) )
```

```
Si IsLeft( arbol ) RecArbolPreOrden( left( arbol ) )
```

```
Si IsRight( arbol ) RecArbolPreOrden( right( arbol ) )
```

```
Func RecArbolPosOrden( arbol )
```

```
Si IsLeft( arbol ) RecArbolPosOrden( left( arbol ) )
```

```
Si IsRight( arbol ) RecArbolPosOrden( right( arbol ) )
```

```
Print( info ( arbol ) )
```

Arbol Estrictamente binario

```
Func ArbolEstrictamenteBinario( arbol )  
Si isLeft( arbol ) and isRight( arbol )  
    Print "por ahora OK"  
    ArbolEstrictamenteBinario( left( arbol ) )  
    ArbolEstrictamenteBinario( right( arbol ) )  
Sino  
    Si Not( isLeft( arbol ) ) and Not( isRight( arbol ) )  
        Print "por ahora OK"  
    Sino  
        Print "El arbol no es estrictamente binario"
```

Ejercicio

Un árbol binario de búsqueda puede implementarse con un array. La representación no enlazada correspondiente consiste en que para cualquier nodo del árbol almacenado en la posición i del array, su hijo izquierdo se encuentra en la posición $2*i$, y su hijo derecho en la posición $2*i+1$. Diseñar a partir de esta representación las funciones con las operaciones correspondientes para gestionar interactivamente un árbol de números enteros.

Maketree()

isLeft() - isRight()