

Reto 4: Árboles

Francisco David Charte Luque

Ignacio Cordón Castillo

1 Inorden no recursivo

Diseñar un procedimiento inorden no recursivo a imagen y semejanza del procedimiento preorden no recursivo que el profesor diseñó en la clase.

```
void inordenNR(const ArbolBinario<int>& a){
    ArbolBinario<int>::Nodo actual;
    stack<ArbolBinario<int>::Nodo> p;

    actual=a.raiz();
    p.push(ArbolBinario<int>::nodo_nulo);
    p.push(actual);

    while (actual!=ArbolBinario<int>::nodo_nulo){
        if (a.izquierda(actual)!=ArbolBinario<int>::nodo_nulo){
            actual= a.izquierda(actual);
            p.push(actual);
        }
        else{
            cout << a.etiqueta(actual) << ' ';
            if (a.derecha(actual)!=ArbolBinario<int>::nodo_nulo){
                actual=a.derecha(actual);
                p.pop();
                p.push(actual);
            }
            else{
                p.pop();
                actual=p.top();

                // Nos vamos al hermano (excepto si se trataba de la raíz)
                if (actual != ArbolBinario<int>::nodo_nulo) {
                    cout << a.etiqueta(actual) << ' ';
                    p.pop();
                    actual = a.derecha(actual);
                }
            }
        }
    }
}
```

```
p.push(actual);
```

```
}  
}  
}  
}  
}
```

2 Codificación de árbol binario

Dar un procedimiento para guardar un árbol binario en disco de forma que se recupere la estructura jerárquica de forma unívoca usando el mínimo número de centinelas que veais posible.