Estructuras de Datos y Algoritmos: Actividad de Repaso
Profesor: Yerko Ortiz

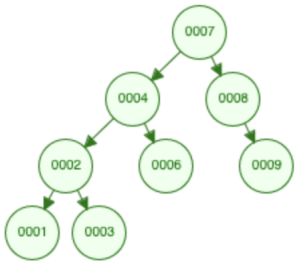
1.

Dada la clase nodo implementada de la siguiente forma:

```
class Node {
    int value;
    Node right, left;
    Node(int value){
        this.value = value;
    }
}
```

Corrija el siguiente algoritmo para un árbol de búsqueda binario. El algoritmo en cuestión es el ancestro mínimo en común, este encuentra el nodo padre en común de dos Nodos dentro de un árbol.
Por ejemplo para el siguiente árbol se puede afirmar lo siguiente:

- El ancestro mínimo en común entre el 1 y 3 es 2.
- El ancestro mínimo en común entre 3 y 6 es 4.
- El ancestro mínimo en común entre 2 y 8 es 7.



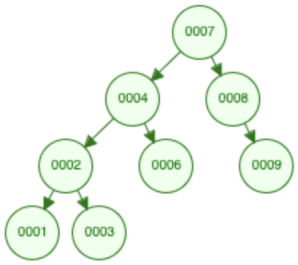

```
    Node lowestCommonAncestor(int v1, int v2){
        Node aux = root;
        while(aux != null){
            if(aux.value > v1 || aux.value > v2) {
                aux = aux.left;
            } else if(aux.value < v1 && aux.value < v2){
                aux = aux.right;
            } else {
                break;
            }
        }
        return aux;
    }
```

1. Encuentre el error en el código y proponga una solución.
2. Analice el código en términos de Big-Oh y proponga una función matemática que caracterice el tiempo de ejecución del código.

2.

Dada la clase nodo implementada de la siguiente forma:

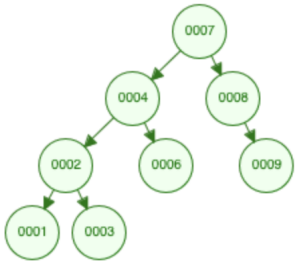
```
class Node {
    int value;
    Node right, left;
    Node(int value){
        this.value = value;
    }
}
```



Sea el siguiente algoritmo misterioso.

```
1       void f(){
2         Deque<Node> q = new ArrayDeque<>();
3         q.add(root);
4         while(!q.isEmpty()){
5             Node aux = q.poll();
6             System.out.println(aux.val);
7
8             if(aux.right != null)
9                 q.add(aux.right);
10
11            if(aux.left != null)
12                q.add(aux.left);
13
14
15        }
16    }
```

1. Ejecute el algoritmo a mano para el árbol de la siguiente figura.



2. Describa su tiempo de ejecución en términos de Big Oh.