Berke Belgin
171044065

3- Firstly in this implementation we can hold head and tail in one variable easily and with this way especially in a single linked data structures while it is taking O(n) to add into queue, it takes O(1) when we hold it in a circular structure since +1 index of tail is head and -1 index vise a versa.

BB

Berke Bolgün
17104065

4-

```
private boolean isLeaf(Node<E> node) {
    if(node.left == null && node.right == null) return true;
    else return false;
}
```

```
public boolean isBalanced () {
    if( isBalanced (root)==-1) return false
    else return true;
}
```

```
public int isBalanced ( Node<E> node) {
    if (node == null ) return 0;
    else {
        int left = isBalanced (node.left).
        int right = isBalanced (node.right);
        if ( left == -1 || right == -1) return -1;
    else if ( left<=right ) return 1+ left + right;
      else {
          if(!isLeaf(node.right)) {
            return -1;
          }
          else return 1+left +right;
      }
}
```

traverse every node once

O(n)

Ω(n)

Θ(n)