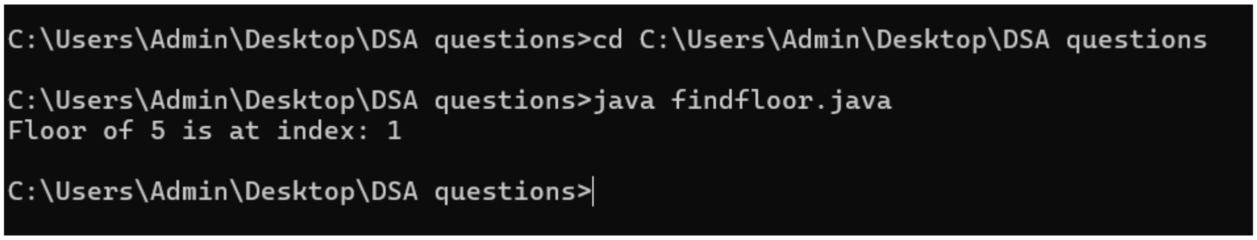
Day 2 practice DSA questions

11/11/2024

1. <u>Floor in sorted array:</u>

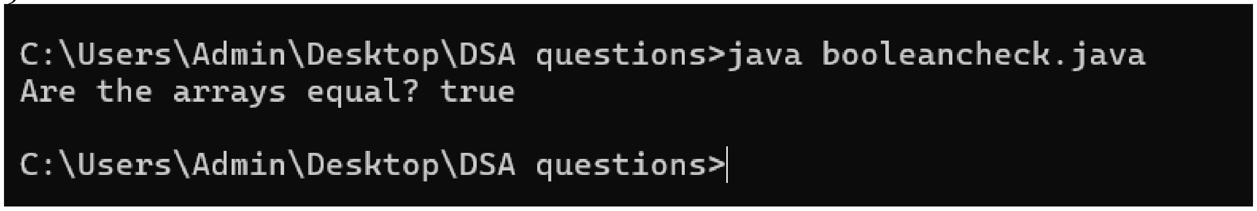
```
class Solution {
    static int findFloor(int[] arr, int k) {
        int low = 0, high = arr.length - 1;
        int ans = -1;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] <= k) {
                ans = mid;
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }


        return ans;

    }

}
```



```
C:\Users\Admin\Desktop\DSA questions>cd C:\Users\Admin\Desktop\DSA questions

C:\Users\Admin\Desktop\DSA questions>java findfloor.java
Floor of 5 is at index: 1

C:\Users\Admin\Desktop\DSA questions>
```

2. Check equal arrays:

```
import java.util.Arrays;

class Solution {

    public static boolean check(int[] arr1, int[] arr2) {
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        if (arr1.length == arr2.length) {
            boolean res = Arrays.equals(arr1, arr2);
            return res;
        }
        return false;
    }

    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3, 4};
        int[] arr2 = {4, 3, 2, 1};

        boolean result = check(arr1, arr2);
        System.out.println("Are the arrays equal? " + result);
    }
}
```



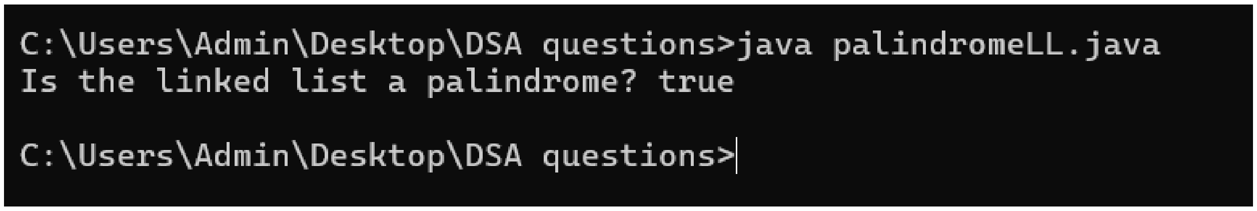
```
C:\Users\Admin\Desktop\DSA questions>java booleancheck.java
Are the arrays equal? true

C:\Users\Admin\Desktop\DSA questions>
```

3.Palindrome linked list:

```
class Solution {
```

```
static class Node {
    int data;
    Node next;

    Node(int data) {
        this.data = data;
        this.next = null;
    }
}

boolean isPalindrome(Node head) {
    if (head == null || head.next == null) {
        return true;
    }

    Node slow = head;
    Node fast = head;

    while (fast != null && fast.next != null) {
        slow = slow.next;
        fast = fast.next.next;
    }

    Node reversedSecondHalf = reverseList(slow);
    Node firstHalf = head;
    Node secondHalf = reversedSecondHalf;

    while (secondHalf != null) {
        if (firstHalf.data != secondHalf.data) {
            return false;
        }
        firstHalf = firstHalf.next;
        secondHalf = secondHalf.next;
    }

    return true;
}

private Node reverseList(Node head) {
    Node prev = null;
    Node current = head;
    while (current != null) {
        Node nextNode = current.next;
        current.next = prev;
        prev = current;
        current = nextNode;
    }
    return prev;
}

public static void main(String[] args) {
    Solution solution = new Solution();

    Node head = new Node(1);
    head.next = new Node(2);
    head.next.next = new Node(2);
    head.next.next.next = new Node(1);

    boolean result = solution.isPalindrome(head);
```

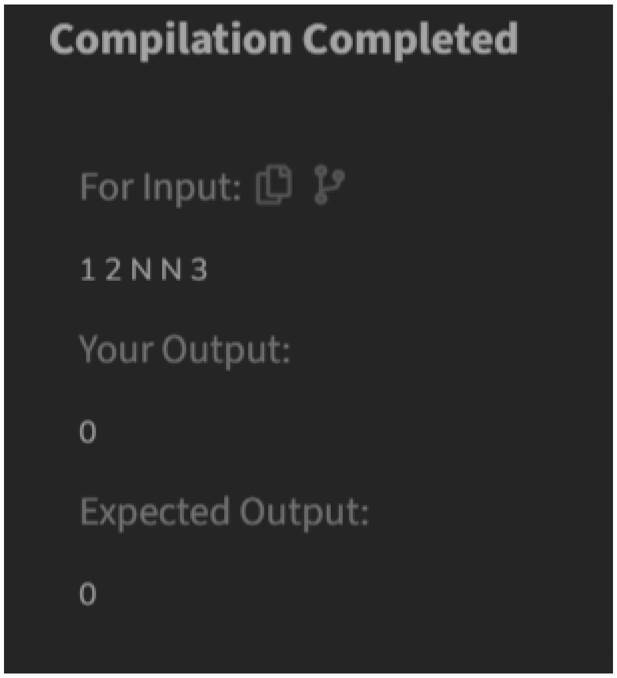
```
        System.out.println("Is the linked list a palindrome? " +
result);
    }
}
```



```
C:\Users\Admin\Desktop\DSA questions>java palindromeLL.java
Is the linked list a palindrome? true

C:\Users\Admin\Desktop\DSA questions>
```

4. <u>Balanced tree check</u>:

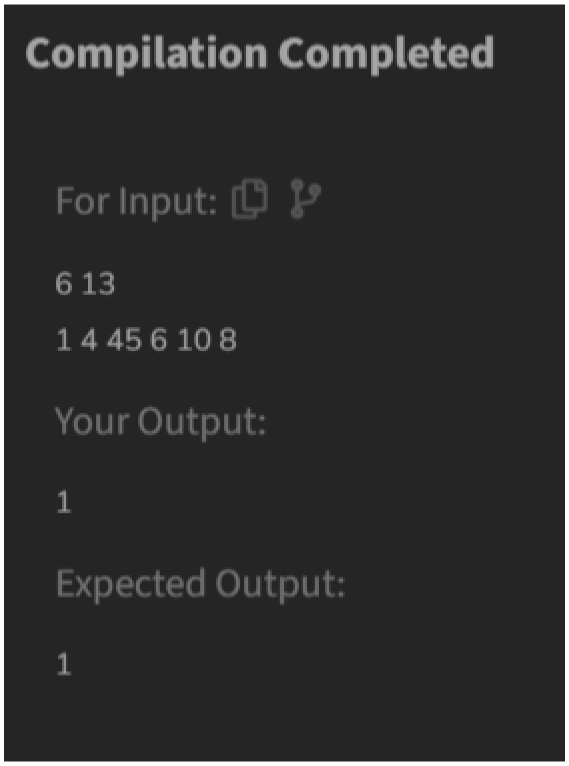
```
class Tree {

    boolean isBalanced(Node root) {
        return height(root) != -1;
    }

    int height(Node root) {
        if (root == null) {
            return 0;
        }

        int leftHeight = height(root.left);
        if (leftHeight == -1) return -1;

        int rightHeight = height(root.right);
        if (rightHeight == -1) return -1;

        if (Math.abs(leftHeight - rightHeight) > 1) {
            return -1;
        }

        return 1 + Math.max(leftHeight, rightHeight);
    }
}
```



```
Compilation Completed

For Input:

1 2 N N 3

Your Output:

0

Expected Output:

0
```

5. Triplet sum in array:

```
class Solution {

    public static boolean find3Numbers(int arr[], int n, int x) {

        Arrays.sort(arr);
        for(int i=0;i<n-2;i++){
            int l=i+1;
            int r=n-1;
            while (l<r){
                int cursum=arr[i]+arr[l]+arr[r];
                if(cursum==x){
                    return true;
                }
                else if (cursum<x){
                    l++;
                }
                else{
                    r--;
                }
            }
        }
        return false;
    }
}
```



```
Compilation Completed

For Input:

6 13
1 4 45 6 10 8

Your Output:

1

Expected Output:

1
```