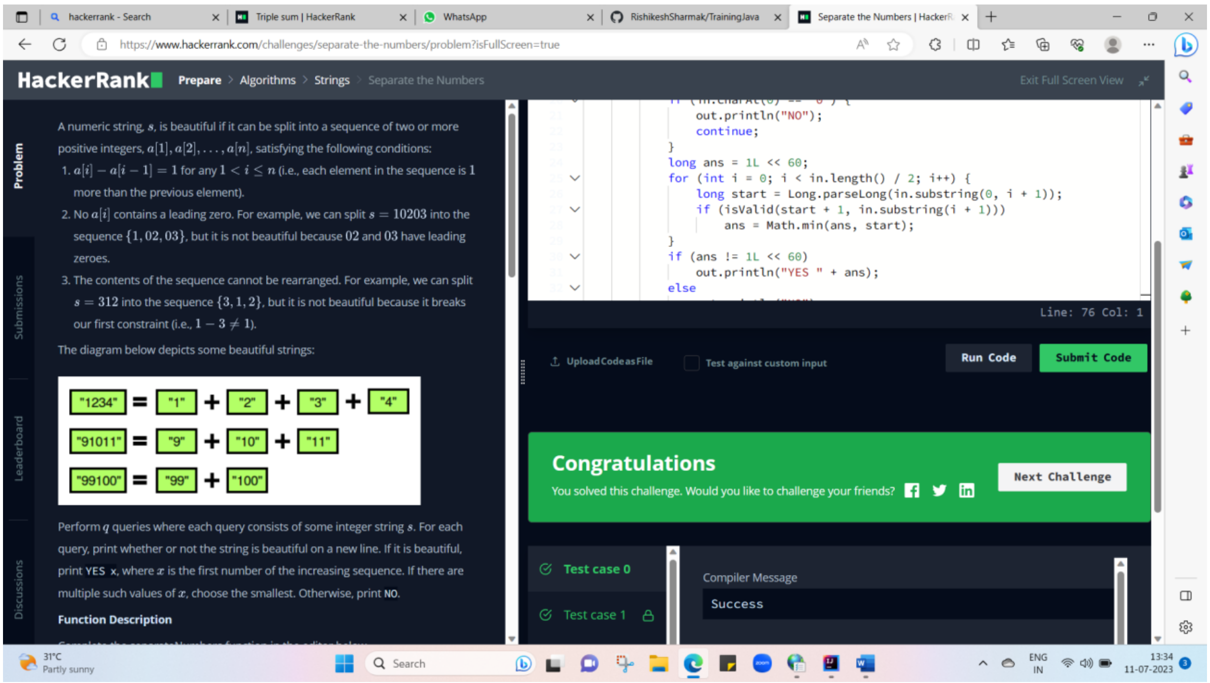
Day 2 Solved HackerRank Problems

1. Separate The Numbers



A numeric string, s, is beautiful if it can be split into a sequence of two or more positive integers, $a[1], a[2], \ldots, a[n]$, satisfying the following conditions:

1. $a[i] - a[i-1] = 1$ for any $1 < i \leq n$ (i.e., each element in the sequence is 1 more than the previous element).
2. No $a[i]$ contains a leading zero. For example, we can split $s = 10203$ into the sequence $\{1, 02, 03\}$, but it is not beautiful because 02 and 03 have leading zeroes.
3. The contents of the sequence cannot be rearranged. For example, we can split $s = 312$ into the sequence $\{3, 1, 2\}$, but it is not beautiful because it breaks our first constraint (i.e., $1 - 3 \neq 1$).

The diagram below depicts some beautiful strings:

"1234" = "1" + "2" + "3" + "4"

"91011" = "9" + "10" + "11"

"99100" = "99" + "100"

Perform q queries where each query consists of some integer string s. For each query, print whether or not the string is beautiful on a new line. If it is beautiful, print `YES x`, where x is the first number of the increasing sequence. If there are multiple such values of x, choose the smallest. Otherwise, print `NO`.

Function Description

```
        if (in.charAt(0) == '0') {
            out.println("NO");
            continue;
        }
        long ans = 1L << 60;
        for (int i = 0; i < in.length() / 2; i++) {
            long start = Long.parseLong(in.substring(0, i + 1));
            if (isValid(start + 1, in.substring(i + 1)))
                ans = Math.min(ans, start);
        }
        if (ans != 1L << 60)
            out.println("YES " + ans);
        else
```

Line: 76 Col: 1

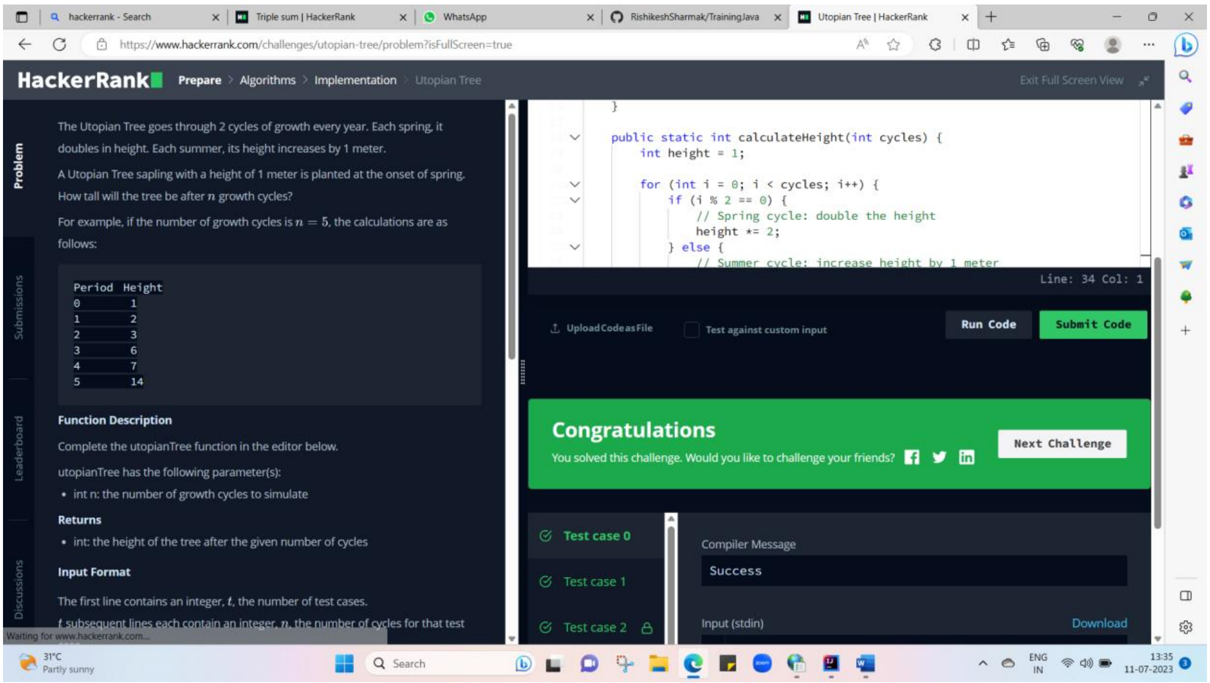
Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

Success

2. Utopian Tree



The Utopian Tree goes through 2 cycles of growth every year. Each spring, it doubles in height. Each summer, its height increases by 1 meter.

A Utopian Tree sapling with a height of 1 meter is planted at the onset of spring. How tall will the tree be after n growth cycles?

For example, if the number of growth cycles is $n = 5$, the calculations are as follows:

```
Period  Height
0          1
1          2
2          3
3          6
4          7
5          14
```

Function Description

Complete the utopianTree function in the editor below.

utopianTree has the following parameter(s):

- int n: the number of growth cycles to simulate

Returns

- int: the height of the tree after the given number of cycles

Input Format

The first line contains an integer, t, the number of test cases.

t subsequent lines each contain an integer, n, the number of cycles for that test

```
    }

    public static int calculateHeight(int cycles) {
        int height = 1;

        for (int i = 0; i < cycles; i++) {
            if (i % 2 == 0) {
                // Spring cycle: double the height
                height *= 2;
            } else {
                // Summer cycle: increase height by 1 meter
```

Line: 34 Col: 1

Congratulations

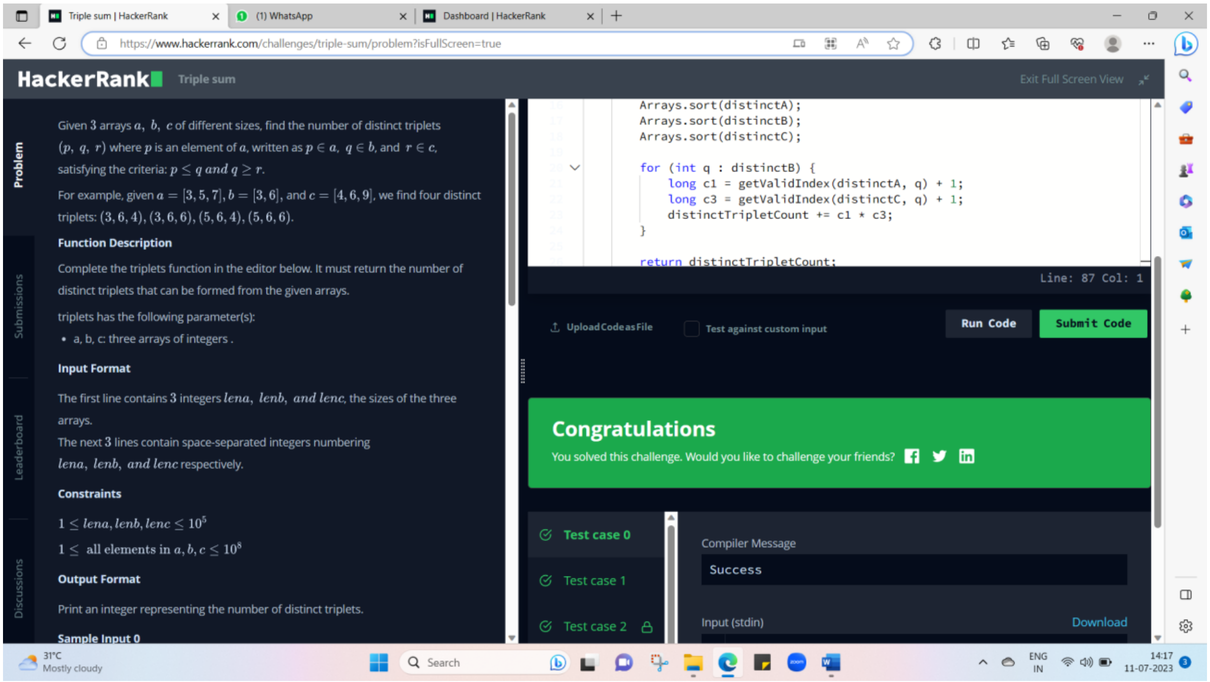
You solved this challenge. Would you like to challenge your friends?

Compiler Message

Success

Input (stdin)

3.Triple Sum



Given 3 arrays $a,\ b,\ c$ of different sizes, find the number of distinct triplets $(p,\ q,\ r)$ where p is an element of a, written as $p \in a,\ q \in b$, and $r \in c$, satisfying the criteria: $p \leq q \ and \ q \geq r$.

For example, given $a = [3,5,7], b = [3,6]$, and $c = [4,6,9]$, we find four distinct triplets: $(3,6,4),(3,6,6),(5,6,4),(5,6,6)$.

Function Description

Complete the triplets function in the editor below. It must return the number of distinct triplets that can be formed from the given arrays.

triplets has the following parameter(s):

- a, b, c: three arrays of integers .

Input Format

The first line contains 3 integers $lena,\ lenb,\ and\ lenc$, the sizes of the three arrays.

The next 3 lines contain space-separated integers numbering $lena,\ lenb,\ and\ lenc$ respectively.

Constraints

$1 \leq lena, lenb, lenc \leq 10^5$

$1 \leq \text{ all elements in } a, b, c \leq 10^8$

Output Format

Print an integer representing the number of distinct triplets.

Sample Input 0

```
        Arrays.sort(distinctA);
        Arrays.sort(distinctB);
        Arrays.sort(distinctC);

        for (int q : distinctB) {
            long c1 = getValidIndex(distinctA, q) + 1;
            long c3 = getValidIndex(distinctC, q) + 1;
            distinctTripletCount += c1 * c3;
        }

        return distinctTripletCount;
```

Line: 87 Col: 1

Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

Input (stdin)