

Day 2 Solved HackerRank Problems

1. Separate The Numbers

HackerRank Prepare > Algorithms > Strings > Separate the Numbers

Problem

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

- $a[i] - a[i - 1] = 1$ for any $1 < i \leq n$ (i.e., each element in the sequence is 1 more than the previous element).
- 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.
- 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

Complete the function `isBeautiful` in the editor below.

Code

```
function isBeautiful(s) {
    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

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [Next Challenge](#)

Test case 0 ✓ Compiler Message

Test case 1 ✓ Success

2. Utopian Tree

HackerRank Prepare > Algorithms > Implementation > Utopian Tree

Problem

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 case.

Code

```
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
            height++;
        }
    }
    return height;
}
```

Line: 34 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [Next Challenge](#)

Test case 0 ✓ Compiler Message

Test case 1 ✓ Success

Test case 2 ✓ Input (stdin) Download

3.Triple Sum

The screenshot shows the HackerRank 'Triple sum' challenge page. The left sidebar contains navigation links: Problem, Submissions, Leaderboard, and Discussions. The main content area on the left details the problem: given three arrays a , b , and c , find the number of distinct triplets (p, q, r) where p is an element of a , q is an element of b , and r is an element of c , satisfying the criteria: $p \leq q$ and $q \geq r$. An example is provided: given $a = [3, 5, 7]$, $b = [3, 6]$, and $c = [4, 6, 9]$, we find four distinct triplets: $(3, 6, 4)$, $(3, 6, 6)$, $(5, 6, 4)$, and $(5, 6, 6)$. The 'Function Description' section states that the `triplets` function must return the number of distinct triplets. The 'Input Format' section describes the input: the first line contains three integers $lena$, $lenb$, and $lenc$, and the next three lines contain space-separated integers for each array. The 'Constraints' section specifies $1 \leq lena, lenb, lenc \leq 10^5$ and $1 \leq \text{all elements in } a, b, c \leq 10^8$. The 'Output Format' section states to print an integer representing the number of distinct triplets. A 'Sample Input 0' is provided at the bottom of the problem description.

The code editor on the right shows the following Java code:

```
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;
```

Below the code editor, there are buttons for 'Upload Code as File', 'Test against custom input', 'Run Code', and 'Submit Code'. A green banner displays 'Congratulations' and 'You solved this challenge. Would you like to challenge your friends?' with social media icons. Below this, a table shows the test results:

Test Case	Status
Test case 0	Success
Test case 1	Success
Test case 2	Success

The 'Compiler Message' section shows 'Success'. The 'Input (stdin)' section is empty. A 'Download' button is located at the bottom right of the input section.