Determining DNA Health ★

19 more points to get your first star!

Rank: 2938780 | Points: 11/30

Problem Solving

Problem Submissions Leaderboard Editorial

DNA is a nucleic acid present in the bodies of living things. Each piece of DNA contains a number of genes, some of which are beneficial and increase the DNA's total health. Each gene has a health value, and the total health of a DNA is the sum of the health values of all the beneficial genes that occur as a substring in the DNA. We represent genes and DNA as non-empty strings of lowercase English alphabetic letters, and the same gene may appear multiple times as a susbtring of a DNA.

Given the following:

- An array of beneficial gene strings, $genes = [g_0, g_1, \ldots, g_{n-1}]$. Note that these gene sequences are not guaranteed to be distinct.
- An array of gene health values, $health = [h_0, h_1, \ldots, h_{n-1}]$, where each h_i is the health value for gene g_i.
- A set of s DNA strands where the definition of each strand has three components, $start$, end, and d, where string d is a DNA for which genes $g_{start}, \ldots, g_{end}$ are healthy.

Find and print the respective total healths of the unhealthiest (minimum total health) and healthiest (maximum total health) strands of DNA as two space-separated values on a single line.

Input Format

The first line contains an integer, n, denoting the total number of genes.
The second line contains n space-separated strings describing the respective values of $g_0, g_1, \ldots, g_{n-1}$ (i.e., the elements of $genes$).
The third line contains n space-separated integers describing the respective values of $h_0, h_1, \ldots, h_{n-1}$ (i.e., the elements of $health$).
The fourth line contains an integer, s, denoting the number of strands of DNA to process.
Each of the s subsequent lines describes a DNA strand in the form `start end d`, denoting that the healthy genes for DNA strand d are $g_{start}, \ldots, g_{end}$ and their respective correlated health values are $h_{start}, \ldots, h_{end}$.

Constraints

- $1 \leq n, s \leq 10^5$
- $0 \leq h_i \leq 10^7$
- $0 \leq first \leq last < n$
- $1 \leq$ the sum of the lengths of all genes and DNA strands $\leq 2 \times 10^6$
- It is guaranteed that each g_i consists of lowercase English alphabetic letters only (i.e., `a` to `z`).

Output Format

Print two space-separated integers describing the respective total health of the unhealthiest and the healthiest strands of DNA.

Sample Input 0

```
6
a b c aa d b
1 2 3 4 5 6
3
1 5 caaab
0 4 xyz
2 4 bcdybc
```

```
0 19
```

Explanation 0

In the diagrams below, the ranges of beneficial genes for a specific DNA on the left are highlighed in green and individual instances of beneficial genes on the right are bolded. The total healths of the $s = 3$ strands are:

1. *d* = caaab, *first* = 1, *last* = 5

indices	0	1	2	3	4	5
genes	**a**	**b**	**c**	**aa**	**d**	**b**
health	1	2	3	4	5	6

gene	caaab	caaab	caaab	caaab	caaab
value	3	4	4	2	6

The total health of caaab is $3 + 4 + 4 + 2 + 6 = 19$.

2. *d* = xyz, *first* = 0, *last* = 4

indices	0	1	2	3	4	5
genes	**a**	**b**	**c**	**aa**	**d**	**b**
health	1	2	3	4	5	6

gene	xyz
value	0

The total health of xyz is 0, because it contains no beneficial genes.

3. *d* = bcdybc, *first* = 2, *last* = 4

indices	0	1	2	3	4	5
genes	**a**	**b**	**c**	**aa**	**d**	**b**
health	1	2	3	4	5	6

gene	bcdybc	bc**d**ybc	bcdybc
value	3	5	3

The total health of bcdybc is $3 + 5 + 3 = 11$.

The unhealthiest DNA strand is xyz with a total health of 0, and the healthiest DNA strand is caaab with a total health of 19. Thus, we print `0 19` as our answer.

Change Theme Language C#

```
using System.CodeDom.Compiler;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Diagnostics.CodeAnalysis;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Reflection;
using System.Runtime.Serialization;
using System.Text.RegularExpressions;
```

```
using System.Text;
using System;



class Solution
{
    public static void Main(string[] args)
    {
        int n = Convert.ToInt32(Console.ReadLine().Trim());

        List<string> genes = Console.ReadLine().TrimEnd().Split(' ').ToList();

        List<int> health = Console.ReadLine().TrimEnd().Split(' ').ToList().Select(healthTemp => Convert.ToInt32(healthTemp)).ToList();

        int s = Convert.ToInt32(Console.ReadLine().Trim());

        for (int sItr = 0; sItr < s; sItr++)
        {
            string[] firstMultipleInput = Console.ReadLine().TrimEnd().Split(' ');

            int first = Convert.ToInt32(firstMultipleInput[0]);

            int last = Convert.ToInt32(firstMultipleInput[1]);

            string d = firstMultipleInput[2];
        }
    }
}

```

Line: 41 Col: 1

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