SOLVE LATER

Akash and too many assignments

Attempted by: **812** / Accuracy: **37%** / Maximum Score: **30** /
319 Votes

Tag(s): Algorithms, Fenwick Tree, Medium

Akash is an *assignment-o-holic* kind of a person, he's crazy about solving assignments. So, to test his skills, his professor has given him an assignment to solve.

In this assignment, Akash is given a **String S** of **length N** consisting of lowercase letters **(a - z)** only. And he has to take care of **2 types** of operations.

- Operation type #1: [**0 index c**] - 0 denotes the operation type #1. This operation expects you to change the character at index **index** to character **c** i.e., **String[index] = c**.
- Operation type #2: [**1 LeftLimit RightLimit K**] - 1 denotes the operation type #2. It expects you to print the **lexicographically K^{th} smallest character** in the sub-string of **String S** starting from **LeftLimit to RightLimit**, inclusively.

Help Akash out in solving this tricky assignment!

Note: The **lexicographic order** means that the words are arranged in a similar fashion as they are presumed to appear in a dictionary. For example: The words a1a2.....ak will be appearing before the words d1d2....dk in the dictionary.

BEST SUBMISSIONS

LANGUAGE: C

TIME (sec) 1.06594

MEMORY (KiB) 102732

by jicun li

VIEW BEST SUBMISSION VIEW ALL SUBMISSION

CONTRIBUTOR

AUTHOR
Akash Sharma

TESTER
Prateek Gupta

THIS PROBLEM WAS ASKED IN

XSEED FOUNDATION FOR LIFE

CHALLENGE NAME
XSEED Hiring Challenge

SOCIAL SHARE

Input format:
The first line contains 2 space separated integers **N** and **Q**, denoting the length of the string and the number of operations respectively.
The next line has a string **S** of length **N**.
The next **Q** lines denote the update/print operations, where 0 and 1 denote the type of operation which has to happen.

Output format:
For each print operation, output the **lexicographically K^{th} smallest character** in the substring of **S** starting from **LeftLimit to RightLimit** if possible, else print "**Out of range**". (Without quotes)

Constraints:
1 <= **Size of the string** <= 10^6
1 <= **Number of queries** <= 10^5
1 <= **LeftLimit, RightLimit, index** <= **N**
c is a lowercase Latin letter only
1 <= **K** <= **N**
S contains only lowercase latin letters **(a to z)**.

Note: Operations are iterative in nature i.e., after each update operation string S gets updated and subsequent operations will performed on the updated string.

SAMPLE INPUT	SAMPLE OUTPUT
5 3 aabbc 1 2 5 3 0 3 a 1 3 3 1	b a

Explanation

In the first print operation, L = 2, R = 5 and K = 3. So, the substring will be String[2:5] = "abbc", so lexicographically the 3rd smallest character will be **b**. (Second one!)

The second operation is update operation, index = 3 and c = 'a', so it will update the 3rd character. After updating the string, it will now be "**aaabc**".

In the third operation, L = 3, R = 3 and K = 1. So the substring will be String[3:3] = a. So lexicographically, 1st smallest character will be **a**.

Time Limit:	1.0 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB
Marking Scheme:	Marks are awarded when all the testcases pass.
Allowed Languages:	C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic

CODE EDITOR

Enter your code or Upload your code as file.

Save C (gcc 5.4.0)

```
1  /*
2  // Sample code to perform I/O:
3
4  scanf("%s", name);                  // Rea
5  printf("Hi, %s.\n", name);          // Wri
6
7  // Warning: Printing unwanted or ill-f
8  */
9
10 // Write your code here
11
```