Practice Compete Jobs Rank Leaderboard bor0s

Dashboard > Functional Programming > Persistent Structures > Minimum Multiple

Badge Progress

Points: 248.51 Rank: 2071

Minimum Multiple

by abhiranjan

Problem Submissions Leaderboard Discussions Editorial

Calculi is Lambda's older brother. Lambda is mischievous and always annoys Calculi by asking silly questions. This time around, Lambda would like to surprise Calculi by asking a challenging and interesting question. To that end, Lambda gives Calculi an array of N integers, $A = \{a_0, a_1, \ldots, a_{N-1}\}$, followed by K queries. Each query is of two types:

- $Q\ l\ r$: Find the minimum positive integer, M, such that each element in subarray $arr[l \ldots r]$ ($\{a_l, a_{l+1}, \ldots, a_r\}$) divides M.
- $U\ idx\ val$: Multiply the value at idx by val. That is $a'_{idx} = a_{idx} \times val$, where a'_{idx} is the updated value.

Your task is to help Calculi tackle this challenge. For each query of type $"Q\ l\ r"$, find the value of M. As this value can be very large, print the M modulo (10^9+7), i.e., $M\%(10^9+7)$. For query of type $"U\ idx\ val"$, update the required element.

Input Format

The first line contains an integer, N, which represents the length of array, A.
In second line, there are N space-separated integers, $a_0, a_1, \ldots, a_{N-1}$, representing the elements of A.
In third line, there is another integer, K, which is the count of queries to follow.
Then follows K lines, each representing a query of one of the types described above.

Constraints

- $1 \le N \le 5 \times 10^4$
- $1 \le a_i \le 100$, where $i \in [0, N-1]$
- $1 \le K \le 5 \times 10^4$
- $0 \le l \le r < N$
- $0 \le idx < N$
- $1 \le val \le 100$

Output Format

For each query of type `Q l r`, print the value of $M\%(10^9+7)$ on a new line.

Sample Input

```
5
2 5 6 1 9
7
Q 0 4
U 1 2
Q 0 2
Q 3 4
Q 2 4
U 3 8
Q 2 3
```

Sample Output

```
90
30
9
18
24
```

Explanation

Query 1 (Q 0 4): Calculi has to find M for (sub)array $A[0\ldots4] = \{2,5,6,1,9\}$ which is 90.
Query 2 (U 1 2): $a'_1 = a_1 \times 2 = 10$. Now updated array is $A = \{2,10,6,1,9\}$.
Query 3 (Q 0 2): M for subarray $A[0\ldots2] = \{2,10,6\}$ is 30.
Query 4 (Q 3 4): M for subarray $A[3\ldots4] = \{1,9\}$ is 9.
Query 5 (Q 2 4): M for subarray $A[2\ldots4] = \{6,1,9\}$ is 18.
Query 6 (U 3 8): Updated array is $A = \{2,10,6,8,9\}$.
Query 7 (Q 2 3): M for subarray $A[2\ldots3] = \{6,8\}$ is 24.

Tested by Wanbo

Submissions: 207
Max Score: 50
Difficulty: Hard

Rate This Challenge:
☆☆☆☆☆

More

If you cannot find your favorite language here, please check out the algorithm domain. We support over 40 popular languages.

Current Buffer (saved locally, editable)

Racket

```
#lang racket
; Enter your code here. Read input from STDIN. Print output to STDOUT
(define (read-list n) (if (= n 0) '() (cons (read) (read-list (- n 1)))))
(define (make-query action l r) (cons action (cons l (cons r '()))))
(define (query-action query) (car query))
(define (query-first query) (car (cdr query)))
(define (query-second query) (car (cdr (cdr query))))
(define (list-range lst start end) (take (drop lst start) (+ 1 (- end start))))
(define modnum (+ 7 (expt 10 9)))

(define (process-query a-list q)
  (cond ((eq? (query-action q) 'Q) (remainder
                                    (apply lcm (list-range a-list (query-first q) (query-second q)))
                                    modnum))
        ((eq? (query-action q) 'U) (let ((index (query-first q)))
                                     (list-set
                                      a-list
                                      index
                                      (* (list-ref a-list index) (query-second q)))))))

(define (repl-queries a-list k)
  (if (eq? k 0) '()
      (letrec ((q (make-query (read) (read) (read)))
               (processed (process-query a-list q)))
        (if (list? processed)
            (repl-queries processed (- k 1))
            (begin
              (displayln processed)
              (repl-queries a-list (- k 1)))))))

(define n (read))
(define a-list (read-list n))
(define k (read))
(define go (repl-queries a-list k))
```

Line: 1 Col: 1

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

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature