Maximum Subarray Sum

We define the following:

- A subarray of array a of length n is a contiguous segment from $a[i]$ through $a[j]$ where $0 \leq i \leq j < n$.
- The sum of an array is the sum of its elements.

Given an n element array of integers, a, and an integer, m, determine the maximum value of the sum of any of its subarrays modulo m.

Example

$a = [1, 2, 3]$

$m = 2$

The following table lists all subarrays and their moduli:

```
                sum     %2
[1]             1       1
[2]             2       0
[3]             3       1
[1,2]           3       1
[2,3]           5       1
[1,2,3]         6       0
```

The maximum modulus is 1.

Function Description

Complete the maximumSum function in the editor below.

maximumSum has the following parameter(s):

- long a[n]: the array to analyze
- long m: the modulo divisor

Returns

- long: the maximum (subarray sum modulo m)

Input Format

The first line contains an integer q, the number of queries to perform.

The next q pairs of lines are as follows:

- The first line contains two space-separated integers n and (long)m, the length of a and the modulo divisor.
- The second line contains n space-separated long integers $a[i]$.

Constraints

- $2 \leq n \leq 10^5$
- $1 \leq m \leq 10^{14}$
- $1 \leq a[i] \leq 10^{18}$
- $2 \leq$ the sum of n over all test cases $\leq 5 \times 10^5$

Sample Input

```
STDIN       Function
-----       --------
1           q = 1
5 7         a[] size n = 5, m = 7
3 3 9 9 5
```

Sample Output

```
6
```

Explanation

The subarrays of array $a = [3, 3, 9, 9, 5]$ and their respective sums modulo $m = 7$ are ranked in order of length and sum in the following list:

1. $[9] \Rightarrow 9 \% 7 = 2$ and $[9] \rightarrow 9 \% 7 = 2$
   $[3] \Rightarrow 3 \% 7 = 3$ and $[3] \rightarrow 3 \% 7 = 3$
   $[5] \Rightarrow 5 \% 7 = 5$
2. $[9, 5] \Rightarrow 14 \% 7 = 0$
   $[9, 9] \Rightarrow 18 \% 7 = 4$
   $[3, 9] \Rightarrow 12 \% 7 = 5$
   $[3, 3] \Rightarrow 6 \% 7 = 6$
3. $[3, 9, 9] \Rightarrow 21 \% 7 = 0$
   $[3, 3, 9] \Rightarrow 15 \% 7 = 1$
   $[9, 9, 5] \Rightarrow 23 \% 7 = 2$
4. $[3, 3, 9, 9] \Rightarrow 24 \% 7 = 3$
   $[3, 9, 9, 5] \Rightarrow 26 \% 7 = 5$
5. $[3, 3, 9, 9, 5] \Rightarrow 29 \% 7 = 1$

The maximum value for $subarray\ sum \% 7$ for any subarray is 6.

Author devuy11
Difficulty Hard
Max Score 65
Submitted By 38690

NEED HELP?
View discussions
View editorial
View top submissions

RESOURCES
Binary Search

RATE THIS CHALLENGE

MORE DETAILS
Download problem statement
Download sample test cases
Suggest Edits

CHOOSE A TRANSLATION
English

Change Theme Language Swift

```swift
import Foundation

/*
 * Complete the 'maximumSum' function below.
 *
 * The function is expected to return a LONG_INTEGER.
 * The function accepts following parameters:
 *  1. LONG_INTEGER_ARRAY a
 *  2. LONG_INTEGER m
 */

func maximumSum(a: [Int], m: Int) -> Int {
    // Write your code here

}

let stdout = ProcessInfo.processInfo.environment["OUTPUT_PATH"]!
FileManager.default.createFile(atPath: stdout, contents: nil, attributes: nil)
let fileHandle = FileHandle(forWritingAtPath: stdout)!

guard let q = Int((readLine()?.trimmingCharacters(in: .whitespacesAndNewlines))!)
else { fatalError("Bad input") }

for qItr in 1...q {
    guard let firstMultipleInputTemp = readLine()?.replacingOccurrences(of: "\\s+$",
with: "", options: .regularExpression) else { fatalError("Bad input") }
    let firstMultipleInput = firstMultipleInputTemp.split(separator: " ").map{ String($0)
 }

    guard let n = Int(firstMultipleInput[0])
    else { fatalError("Bad input") }

    guard let m = Int(firstMultipleInput[1])
    else { fatalError("Bad input") }

    guard let aTemp = readLine()?.replacingOccurrences(of: "\\s+$", with: "", options:
.regularExpression) else { fatalError("Bad input") }

    let a: [Int] = aTemp.split(separator: " ").map {
        if let aItem = Int($0) {
            return aItem
        } else { fatalError("Bad input") }
    }

    guard a.count == n else { fatalError("Bad input") }

    let result = maximumSum(a: a, m: m)

    fileHandle.write(String(result).data(using: .utf8)!)
    fileHandle.write("\n".data(using: .utf8)!)
}

```

Line: 49 Col: 1

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