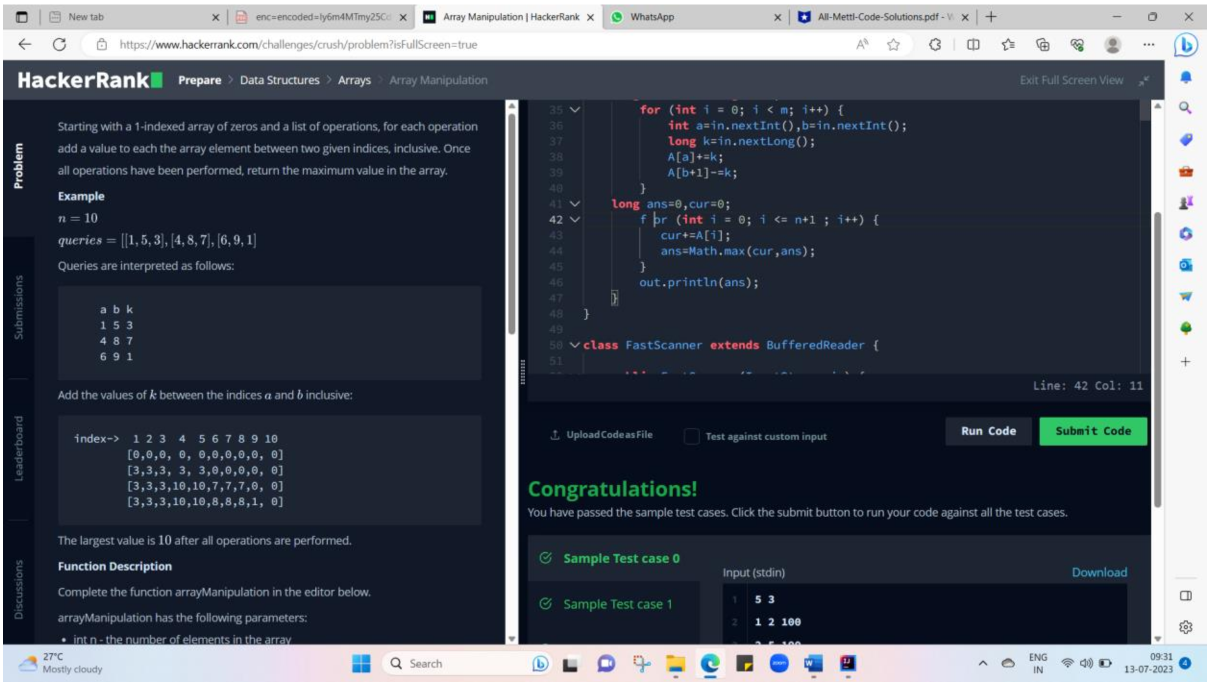
DAY 4 HackerRank solved problem

Array Manipulation


New tab
enc=encoded=ly6m4MTmy25Cc
Array Manipulation | HackerRank
WhatsApp
All-Mettl-Code-Solutions.pdf - W
https://www.hackerrank.com/challenges/crush/problem?isFullScreen=true
HackerRank
Prepare > Data Structures > Arrays > Array Manipulation
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
Line: 42 Col: 11
Upload Code as File
□ Test against custom input
Run Code
Submit Code
Download
27°C
Mostly cloudy
Search
ENG
IN
09:31
13-07-2023

Starting with a 1-indexed array of zeros and a list of operations, for each operation add a value to each the array element between two given indices, inclusive. Once all operations have been performed, return the maximum value in the array.

Example

$n = 10$

$queries = [[1,5,3],[4,8,7],[6,9,1]]$

Queries are interpreted as follows:

```
a b k
1 5 3
4 8 7
6 9 1
```

Add the values of k between the indices a and b inclusive:

```
index->  1 2 3  4  5 6 7 8 9 10
        [0,0,0, 0, 0,0,0,0,0, 0]
        [3,3,3, 3, 3,0,0,0,0, 0]
        [3,3,3,10,10,7,7,7,0, 0]
        [3,3,3,10,10,8,8,8,1, 0]
```

The largest value is 10 after all operations are performed.

Function Description

Complete the function arrayManipulation in the editor below.

arrayManipulation has the following parameters:

- int n - the number of elements in the array

```
        for (int i = 0; i < m; i++) {
            int a=in.nextInt(),b=in.nextInt();
            long k=in.nextLong();
            A[a]+=k;
            A[b+1]-=k;
        }
    long ans=0,cur=0;
        for (int i = 0; i <= n+1 ; i++) {
          cur+=A[i];
          ans=Math.max(cur,ans);
        }
        out.println(ans);
    }
}

class FastScanner extends BufferedReader {
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ Sample Test case 1

Input (stdin)

```
5 3
1 2 100
```

code:

```
import java.io.InputStreamReader;

import java.io.IOException;

import java.util.InputMismatchException;

import java.io.PrintStream;

import java.io.BufferedReader;

import java.io.OutputStream;

import java.io.PrintWriter;

import java.io.Reader;
import java.io.Writer;
import java.io.InputStream;

/**
 * Built using CHelper plug-in
 * Actual solution is at the top
 * @author Nipuna Samarasekara
 */
public class Solution {
    public static void main(String[] args) {
            InputStream inputStream = System.in;
            OutputStream outputStream = System.out;
            FastScanner in = new FastScanner(inputStream);
            FastPrinter out = new FastPrinter(outputStream);
            Task2 solver = new Task2();
            solver.solve(1, in, out);
            out.close();
    }
}

class Task2 {
   /////////////////////////////////////////////////////////////
   public void solve(int testNumber, FastScanner in, FastPrinter
out) {
   int n=in.nextInt(),m=in.nextInt();
   long[] A= new long[n+2];
      for (int i = 0; i < m; i++) {
         int a=in.nextInt(),b=in.nextInt();
         long k=in.nextLong();
         A[a]+=k;
         A[b+1]-=k;
      }
   long ans=0,cur=0;
      for (int i = 0; i <= n+1 ; i++) {
        cur+=A[i];
        ans=Math.max(cur,ans);
      }
      out.println(ans);
   }
}

class FastScanner extends BufferedReader {

   public FastScanner(InputStream is) {
      super(new InputStreamReader(is));
   }
```

```
    public int read() {
        try {
            int ret = super.read();
//            if (isEOF && ret < 0) {
//                throw new InputMismatchException();
//            }
//            isEOF = ret == -1;
            return ret;
        } catch (IOException e) {
            throw new InputMismatchException();
        }
    }

    public String next() {
        StringBuilder sb = new StringBuilder();
        int c = read();
        while (isWhiteSpace(c)) {
            c = read();
        }
        if (c < 0) {
            return null;
        }
        while (c >= 0 && !isWhiteSpace(c)) {
```

```
        sb.appendCodePoint(c);
        c = read();
    }
    return sb.toString();
}

static boolean isWhiteSpace(int c) {
    return c >= 0 && c <= 32;
}

public int nextInt() {
    int c = read();
    while (isWhiteSpace(c)) {
        c = read();
    }
    int sgn = 1;
    if (c == '-') {
        sgn = -1;
        c = read();
    }
    int ret = 0;
    while (c >= 0 && !isWhiteSpace(c)) {
        if (c < '0' || c > '9') {
            throw new NumberFormatException("digit expected " +
(char) c
                  + " found");
         }
         ret = ret * 10 + c - '0';
         c = read();
      }
      return ret * sgn;
   }

   public long nextLong() {
      return Long.parseLong(next());
   }

   public String readLine() {
      try {
         return super.readLine();
      } catch (IOException e) {
         return null;
      }
   }

   }
```

```
class FastPrinter extends PrintWriter {

    public FastPrinter(OutputStream out) {
        super(out);
    }

    public FastPrinter(Writer out) {
        super(out);
    }


}
```