Distributed Online Round

A. Testrun
B. almost_sorted
C. mutexes
D. johnny
E. highest_mountain

Contest Analysis
Questions asked 6

Submissions

Testrun
0pt | Not attempted **0/64 users** correct (0%)

almost_sorted
1pt | Not attempted **194/203 users** correct (96%)
7pt | Not attempted **104/187 users** correct (56%)

mutexes
2pt | Not attempted **84/147 users** correct (57%)
20pt | Not attempted **48/69 users** correct (70%)

johnny
2pt | Not attempted **91/105 users** correct (87%)
30pt | Not attempted **17/70 users** correct (24%)

highest_mountain
1pt | Not attempted **43/61 users** correct (70%)
37pt | Not attempted **0/9 users** correct (0%)

Top Scores

mk.al13n	63
ecnerwala	63
shik	63
Marcin.Smulewicz	56
WJMZBMR	56
bmerry	43
ZbanIlya	43
wan92hy	42
simonlindholm	42
dreamoon	42

Problem B. almost_sorted

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 1 points *2 minute timeout*	The contest is finished.
large 7 points *10 minute timeout*	The contest is finished.

Don't know what distributed problems are about? See our guide.

Problem

As a *very important director* of a *very important company*, you have a lot of files to keep track of. You do this by carefully keeping them sorted. However, you took a well-deserved vacation last week, and when you came back, you discovered to your horror that someone has put the files out of place!

They are not very much out of place, in fact they're still *almost* sorted. More precisely, the file that should be in position i if the files were sorted is now at most K positions away — that is, somewhere between position $i - K$ and $i + K$, inclusive.
However, you can't work like this. So, you ask your assistants to put the files into their correct places. Each file has an identifier, and files with a larger identifier should be placed after those with the smaller identifier. They will not change the relative order of files with the same identifier.
To verify the files are sorted correctly, you will ask your assistants to calculate a simple checksum — for each file multiply that file's identifier by its position (beginning from 0), and sum this for all files, modulo 2^{20}.

Unfortunately, to make use of the checksum, you have to know what it's value should be. So, write a program that will output the expected checksum after sorting the files.

Input

The input library will be called "almost_sorted"; see the sample inputs below for examples in your language. It will define three methods: NumberOfFiles(), which will return the number of files, MaxDistance() — the maximum difference between the current and desired position of any file, and Identifier(i), which will return the value of the identifier of the file that's currently standing on position i, for $0 \le i <$ NumberOfFiles().
A single call to Identifier() will take approximately 0.04 microseconds.

Output

Output one number — the value of the checksum for the sorted sequence of files, modulo 2^{20}.

Limits

Each node will have access to 128MB of RAM, and a time limit of 3 seconds.
$0 \le$ Identifier(i) $\le 10^{18}$, for $0 \le i <$ NumberOfFiles().

Small input

Your solution will run on 10 nodes.
$0 \le$ MaxDistance() $<$ NumberOfFiles() ≤ 1000.

Large input

Your solution will run on 100 nodes.
$1 \le$ NumberOfFiles() $\le 10^8$.
$0 \le$ MaxDistance() $<$ NumberOfFiles().
$0 \le$ MaxDistance() $\le 10^6$.

Sample

Input	Output
See sample input files below.	For sample input 1: 0 For sample input 2: 7000 For sample input 3: 154112

Sample input libraries:
Sample input for test 1: almost_sorted.h [CPP] almost_sorted.java [Java]
Sample input for test 2: almost_sorted.h [CPP] almost_sorted.java [Java]
Sample input for test 3: almost_sorted.h [CPP] almost_sorted.java [Java]

All problem statements, input data and contest analyses are licensed under the Creative Commons Attribution License.

© 2008-2017 Google Google Home - Terms and Conditions - Privacy Policies and Principles

Powered by

Google Cloud Platform