Боштук

N, 0дөн чоң же барабар сан бар $0 \leq a_1 < a_2 < \cdots < a_N \leq 10^{18}$. 1 ден $N - 1$ге чейинки бардык i үчүн $a_{i+1} - a_i$ нин *эң чоң* маанисин тап. Бирок сандар берилбейт, сандарга функция аркылуу байланышсак болот. (грайдер)

Implementation for C and C++ (Грайдер, функциялар)

`findGap(T, N)` функциясын жазыш керек, `long long` түрүндө эки сан:

- `T` — subtaskтын номери (1 or 2)
- `N` — эррейдин саны

`findGap` фунциянын ичинде `MinMax(s, t, &mn, &mx)` функциясын колдонсон болот.
`MinMax(s, t, &mn, &mx)` биринчи эки параметр `s` жана `t` - `long long` жана кийинки эки параметр `&mn` жана `&mx` pointerлер `long long`.
`MinMax(s, t, &mn, &mx)` функцияны чакырганда `mn`дин мааниси `s`тен чоң же барабар болгон эң кичине a_iге өзгөрөт `mx`тин мааниси `t`ден кичине же барабар болгон эң чоң a_jга өзгөрөт, эгерде бул шарттарга туура келген сан болбосо `s` жана `t` (inclusive) лар `-1`ге барабар болуп калат.
`s` <= `t` болуусу шарт.

Implementation for all

In addition to the standard requirements (time and memory limits, no runtime errors, etc), your submission has to achieve the following in order to solve a testcase:

- your function `findGap` must return the correct answer,
- the cost M associated with calls to function `MinMax` must not exceed the allowed limit (see section Scoring).

Example for C, C++

Consider the case where $N = 4$ and $a_1 = 2, a_2 = 3, a_3 = 6$, and $a_4 = 8$.

The answer, which is 3, can be calculated and thus returned by `findGap` if the following calls to `MinMax` are made:

- `MinMax(1, 2, &mn, &mx)` is called and `mn` and `mx` both have the value 2.
- `MinMax(3, 7, &mn, &mx)` is called and `mn` have the value 3 and `mx` has the value 6.
- `MinMax(8, 9, &mn, &mx)` is called and `mn` and `mx` both have the value 8.

Scoring

In all subtasks the constraint $2 \leq N \leq 100,000$ holds.

Subtask 1 (30 points): Each call to `MinMax` will add 1 to M. You will receive the full score for the subtask if $M \leq \frac{N+1}{2}$ for all test cases.

Subtask 2 (70 points): Let k be the number of input integers larger than or equal to `s` and smaller than or equal to `t` in a call to `MinMax`. Each call to `MinMax` will add $k+1$ to M. The final score will be calculated by the following rule: Final score for the subtask is the minimum score you received among all test cases. For a test case, the score is 70 if $M \leq 3N$ and the score is $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$, otherwise.

Experimentation

The sample grader which can be downloaded from the scoring system will read data from standard input. The first line of input should contain two integers, subtask number T, and N. The next line should contain N integers in ascending order. The sample grader will write to standard output the value returned by `findGap` in the first line and the value of M appropriate for the subtask the input test case belongs to.

The following input describes the above example:

```
2 4
2 3 6 8
```