



## Промежуток

Есть  $N$  неотрицательных целых чисел  $a_1, a_2, \dots, a_N$ , удовлетворяющих следующему неравенству  $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$ . Жихак хочет найти *максимальное возможное* значение  $a_{i+1} - a_i$  где  $i$  от 1 до  $N - 1$ . Эти числа не будут даны программе напрямую, но информацию о них можно будет получать с помощью специальной функции. Подробности указаны в секции "Детали реализации" выбранного вами языка.

## Постановка задачи

Помогите Жихаку реализовать функцию, которая вернет *максимальное возможное* значение  $a_{i+1} - a_i$  где  $i$  от 1 до  $N - 1$ .

## Детали реализации для C и C++

Вам необходимо реализовать функцию `findGap(T, N)` которая принимает следующие параметры и возвращает целое число типа `long long`:

- $T$  — номер подзадачи (1 или 2)
- $N$  — количество чисел

Функция `findGap` может вызывать функцию `MinMax(s, t, &mn, &mx)` у которой первые два параметра  $s$  и  $t$  это целые числа типа `long long`, а последние два параметра `&mn` и `&mx` это указатели на переменные типа `long long`, то есть, `mn` и `mx` переменные типа `long long`. После завершения `MinMax(s, t, &mn, &mx)`, в переменной `mn` будет записано минимальное  $a_i$ , не меньшее значения  $s$ , а в переменной `mx` будет записано наибольшее значение  $a_j$ , не большее  $t$ . Если между  $s$  и  $t$  (включительно) не находится ни одного из чисел  $a_i$ , то и `mn` и `mx` будут иметь значение `-1`. Значение  $s$  должно быть не больше, чем значение  $t$ , во всех вызовах функции `MinMax`. Если это условие не будет выполнено, то программа завершится с ненулевым кодом возврата.

## Детали реализации для Pascal

Вам необходимо реализовать функцию `findGap(T, N)` которая принимает следующие параметры и возвращает целое число типа `Int64`:

- $T$  — номер подзадачи (1 или 2) (типа `Integer`)
- $N$  — количество чисел (типа `LongInt`)

Функция `findGap` может вызывать функцию `MinMax(s, t, mn, mx)` у которой первые два параметра  $s$  и  $t$  это целые числа типа `Int64`, а последние два параметра `mn` и `mx` передаются по ссылке и имеют тип `Int64`. После завершения `MinMax(s, t, mn, mx)`, в переменной `mn` будет записано минимальное  $a_i$ , не меньшее значения  $s$ , а в переменной `mx` будет записано наибольшее значение  $a_j$ , не большее  $t$ . Если между  $s$  и  $t$  (включительно) не находится ни одного из чисел  $a_i$ , то и `mn` и `mx` будут иметь значение `-1`. Значение  $s$  должно быть не больше,

чем значение  $t$  во всех вызовах функции `MinMax`. Если это условие не будет выполнено, то программа будет завершена.

## Детали реализации для всех языков

В дополнение к обычным требованиям (ограничения по времени и памяти, отсутствие ошибок времени выполнения, и.т.д.), ваша посылка должна удовлетворять следующим ограничениям, чтобы тест был засчитан:

- функция `findGap` должна вернуть правильный ответ,
- стоимость  $M$  вызовов функции `MinMax` не должна превосходить определённого значения (см. секцию Система Оценивания).

## Пример для C, C++

Пусть  $N = 4$ ,  $a_1 = 2$ ,  $a_2 = 3$ ,  $a_3 = 6$  и  $a_4 = 8$ .

Ответ, равный **3**, может быть вычислен функцией `findGap`, если сделать следующие вызовы функции `MinMax`:

- `MinMax(1, 2, &mn, &mx)`, после чего `mn` и `mx` равны **2**.
- `MinMax(3, 7, &mn, &mx)`, после чего `mn` равен **3**, `mx` равен **6**.
- `MinMax(8, 9, &mn, &mx)`, после чего `mn` и `mx` равны **8**.

## Пример для Pascal

Пусть  $N = 4$ ,  $a_1 = 2$ ,  $a_2 = 3$ ,  $a_3 = 6$  и  $a_4 = 8$ .

Ответ, равный **3**, может быть вычислен функцией `findGap`, если сделать следующие вызовы функции `MinMax`:

- `MinMax(1, 2, mn, mx)`, после чего `mn` и `mx` равны **2**.
- `MinMax(3, 7, mn, mx)`, после чего `mn` равен **3**, `mx` равен **6**.
- `MinMax(8, 9, mn, mx)`, после чего `mn` и `mx` равны **8**.

## Система оценивания

Во всех подзадачах  $2 \leq N \leq 100,000$ .

**Подзадача 1 (30 баллов):** Каждый вызов `MinMax` прибавляет **1** к  $M$ . Вы получите полный балл за подзадачу, если если  $M \leq \frac{N+1}{2}$  во всех тестах.

**Подзадача 2 (70 баллов):** Пусть  $k$  -- количество чисел из  $a$  между  $s$  и  $t$  включительно в вызове функции `MinMax`. Каждый вызов `MinMax` прибавит  $k + 1$  к  $M$ . Балл за подзадачу будет вычислен как минимальный из баллов за тесты этой подзадачи. Балл за тест равен 70, если  $M \leq 3N$ , и  $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ , в противном случае.

## Пример проверяющего модуля

Пример проверяющего модуля, который можно скачать в тестирующей системе, читает входные данные со стандартного ввода. Первая строка входных данных должна содержать два числа - номер подзадачи и  $T$ , и  $N$ . Следующая строка должна содержать  $N$  чисел в возрастающем порядке. Пример проверяющего модуля выведет на стандартный вывод значение, которое вернула функция `findGap`, и значение  $M$  в соответствии с подзадачей, указанной во вводе.

Пример описывается следующим вводом:

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