



最大差分

问题描述

有 N 个严格递增的非负整数 a_1, a_2, \dots, a_N ($0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$)。你需要找出 $a_{i+1} - a_i$ ($0 \leq i \leq N - 1$) 里最大的值。

你的程序不能直接读入这个整数序列，但是你可以通过给定的函数来查询该序列的信息。关于查询函数的细节，请根据你所使用的语言，参考下面的**实现细节**部分。

你需要实现一个函数，该函数返回 $a_{i+1} - a_i$ ($0 \leq i \leq N - 1$) 中的最大值。

实现细节 (C/C++)

你需要实现一个函数 `findGap(T, N)`，该函数接受下面的参数，并返回一个 `long long` 类型的整数：

- `T` — 子任务的编号 (1或者2)
- `N` — 序列的长度

你的函数 `findGap` 可以调用系统提供的查询函数 `MinMax(s, t, &mn, &mx)`，该函数的前两个参数 `s` 和 `t` 是 `long long` 类型的整数，后两个参数 `&mn` 和 `&mx` 是 `long long` 类型的整数的指针 (`mn` 和 `mx` 是 `long long` 类型的整数)。当 `MinMax(s, t, &mn, &mx)` 返回时，变量 `mn` 将会存储满足 $i \in [s, t]$ 中 a_i 的最小值，变量 `mx` 将会存储满足 $i \in [s, t]$ ， a_i 的最大值。如果区间 $[s, t]$ 中没有序列中的数，则 `mn` 和 `mx` 都将存储 -1。在查询时需要满足 $s \leq t$ ，否则程序将会终止，该测试点计为 0 分。

实现细节 (Pascal)

你需要实现一个函数 `findGap(T, N)`，该函数接受下面的参数，并返回一个 `Int64` 类型的整数：

- `T` — 子任务的编号 (1或者2) (`Integer` 类型)
- `N` — 序列的长度 (`LongInt` 类型)

你的函数 `findGap` 可以调用系统提供的查询过程 `MinMax(s, t, mn, mx)`，该过程的前两个参数 `s` 和 `t` 是 `Int64` 类型的整数，后两个参数 `mn` 和 `mx` 是传引用方式的 `Int64` 类型的整数 (过程内部对这两个变量的修改会影响到外部的对应变量的值)。当 `MinMax(s, t, mn, mx)` 执行完毕时，变量 `mn` 将会存储满足 $i \in [s, t]$ ， a_i 的最小值，变量 `mx` 将会存储满足 $i \in [s, t]$ ， a_i 的最大值。如果区间 $[s, t]$ 中没有序列中的数，则 `mn` 和 `mx` 都将存储 -1。在查询时需要满足 $s \leq t$ ，否则程序将会终止，该测试点计为 0 分。

实现细节 (所有语言)

为了得到每个测试点的分数，除了需要满足标准的需求外 (时间和空间限制，没有运行错误

等)，你的程序还需要满足下面两个要求：

- 你的函数findGap必须返回正确的答案，
- 花费的代价M不能超出给定的限制（关于M的定义参考得分部分）。

样例 C, C++

考虑 $N = 4, a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$ 。

则答案应该是3，可以通过下面的几组对MinMax的询问获得：

- 调用MinMax(1, 2, &mn, &mx)，则mn和mx皆返回2。
- 调用MinMax(3, 7, &mn, &mx)，则mn返回3，mx返回6。
- 调用MinMax(8, 9, &mn, &mx)，则mn和mx皆返回8。

样例 Pascal

考虑 $N = 4, a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$ 。

则答案应该是3，可以通过下面的几组对MinMax的询问获得：

- 调用MinMax(1, 2, mn, mx)，则mn和mx皆返回2。
- 调用MinMax(3, 7, mn, mx)，则mn返回3，mx返回6。
- 调用MinMax(8, 9, mn, mx)，则mn和mx皆返回8。

子任务及部分分

对所有的测试点，有 $2 \leq N \leq 100,000$ 。

每一个测试点开始测试之前， M 都将被初始化为0。

子任务 1 (30 分)：每一次调用MinMax都将使 M 加1。为了获得所有分数，需要满足：对于该子任务下的所有测试点，都有 $M \leq \frac{N+1}{2}$ 。

子任务 2 (70 分)：定义 k 为调用MinMax时，区间 $[s, t]$ 中的序列中数的数量。每次调用MinMax，将使 M 加上 $k + 1$ 。对于每一个测试点，如果 $M \leq 3N$ ，你将得到70分，否则将得到 $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ 分。你的该子任务的得分是其下所有测试点中的最低分。

计分程序

你可以从系统上下载计分程序进行自我测试。该计分程序从标准输入中读入两行。第一行包含两个整数，子任务编号 T ，和序列长度 N 。第二行包含 N 个严格递增的非负整数。然后该程序会向标准输出中写入两行，第一行为findGap的返回值，第二行为花费 M 的值。

下面的输入描述了上面的样例：

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