



Celah

Terdapat N buah bilangan bulat non-negatif a_1, a_2, \dots, a_N yang memenuhi pertidaksamaan $0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$. Jeehak ingin menemukan nilai *tertinggi yang mungkin* dari $a_{i+1} - a_i$ yang mana nilai i berkisar dari 1 hingga $N - 1$. Bilangan-bilangan pada masukan tidak akan diberikan langsung ke program Jeehak, melainkan dapat diperoleh dari pemanggilan suatu fungsi khusus. Lihat bagian Implementasi dalam bahasa pemrograman pilihan Anda untuk keterangan lebih lanjut.

Soal

Bantulah Jeehak untuk mengimplementasikan sebuah fungsi yang mengembalikan nilai tertinggi yang mungkin dari $a_{i+1} - a_i$ yang mana nilai i berkisar dari 1 hingga $N - 1$.

Implementasi dalam C dan C++

Anda diminta untuk mengimplementasikan sebuah fungsi `findGap(T, N)` yang menerima parameter-parameter berikut dan mengembalikan sebuah bilangan bulat dengan tipe data `long long`:

- T — nomor subsoal (1 atau 2)
- N — banyaknya bilangan bulat yang diberikan

Fungsi `findGap` Anda dapat memanggil fungsi `MinMax(s, t, &mn, &mx)` dimana dua parameter pertama s dan t merupakan variabel dengan tipe data `long long` serta dua parameter terakhir `&mn` dan `&mx` merupakan pointer untuk variabel dengan tipe data `long long`, dengan kata lain, mn dan mx merupakan variabel dengan tipe data `long long`. Ketika fungsi `MinMax(s, t, &mn, &mx)` dikembalikan, variabel mn akan berisi nilai a_i terkecil yang lebih besar atau sama dengan nilai s dan variabel mx akan berisi nilai a_j terbesar yang lebih kecil atau sama dengan nilai t . Pada kasus tidak ada satu pun bilangan-bilangan pada masukan yang berada di antara s dan t (inklusif), maka kedua variabel mn dan mx akan berisi nilai -1 . Nilai s tidak boleh lebih besar dari nilai t ketika `MinMax` dipanggil. Jika kondisi ini tidak dipenuhi, program akan dihentikan dengan sebuah *non-zero exit code*.

Implementasi dalam Pascal

Anda diminta untuk mengimplementasikan sebuah fungsi `findGap(T, N)` yang menerima parameter-parameter berikut dan mengembalikan sebuah bilangan bulat dengan tipe data `Int64`:

- T — nomor subsoal (1 atau 2) (dengan tipe data `Integer`)
- N — banyaknya bilangan bulat yang diberikan (dengan tipe data `LongInt`)

Fungsi `findGap` Anda dapat memanggil fungsi `MinMax(s, t, mn, mx)` dimana dua parameter pertama s dan t merupakan variabel dengan tipe data `Int64` serta dua parameter terakhir mn dan mx merupakan **parameter acuan (by reference)** dari variabel dengan tipe data `Int64`, dengan kata lain, mn dan mx merupakan variabel dengan tipe data `Int64`. Ketika fungsi `MinMax(s, t, mn, mx)` selesai, variabel mn akan berisi nilai a_i terkecil yang lebih besar atau sama dengan nilai s dan variabel

mx akan berisi nilai a_j terbesar yang lebih kecil atau sama dengan nilai t . Pada kasus tidak ada satu pun bilangan-bilangan pada masukan yang berada di antara s dan t (inklusif), maka kedua variabel mn dan mx akan berisi nilai -1 . Nilai s tidak boleh lebih besar dari nilai t ketika `MinMax` dipanggil. Jika kondisi ini tidak dipenuhi, program akan dihentikan.

Implementasi untuk semua

Selain dari syarat-syarat standar (batasan waktu dan memori, tidak *runtime error*, dll), program yang Anda kumpulkan juga harus memenuhi syarat-syarat tambahan berikut untuk dapat menyelesaikan sebuah kasus uji:

- fungsi `findGap` Anda harus mengembalikan jawaban yang benar,
- nilai M yang berhubungan dengan pemanggilan fungsi `MinMax` tidak boleh melebihi batasan yang ditentukan (lihat bagian Penilaian).

Contoh untuk C dan C++

Tinjau kasus dimana nilai $N = 4$ serta $a_1 = 2, a_2 = 3, a_3 = 6$, dan $a_4 = 8$.

Jawaban untuk kasus tersebut, yakni **3**, dapat dihitung dan dikembalikan oleh fungsi `findGap` jika pemanggilan fungsi `MinMax` berikut dilakukan:

- `MinMax(1, 2, &mn, &mx)` dipanggil sehingga mn dan mx keduanya berisi nilai **2**.
- `MinMax(3, 7, &mn, &mx)` dipanggil sehingga mn berisi nilai **3** dan mx berisi nilai **6**.
- `MinMax(8, 9, &mn, &mx)` dipanggil sehingga mn dan mx keduanya berisi nilai **8**.

Contoh untuk Pascal

Tinjau kasus dimana nilai $N = 4$ serta $a_1 = 2, a_2 = 3, a_3 = 6$, dan $a_4 = 8$.

Jawaban untuk kasus tersebut, yakni **3**, dapat dihitung dan dikembalikan oleh fungsi `findGap` jika pemanggilan fungsi `MinMax` berikut dilakukan:

- `MinMax(1, 2, mn, mx)` dipanggil sehingga mn dan mx keduanya berisi nilai **2**.
- `MinMax(3, 7, mn, mx)` dipanggil sehingga mn berisi nilai **3** dan mx berisi nilai **6**.
- `MinMax(8, 9, mn, mx)` dipanggil sehingga mn dan mx keduanya berisi nilai **8**.

Penilaian

Untuk semua subsoal, berlaku $2 \leq N \leq 100,000$.

Subsoal 1 (30 poin): Setiap pemanggilan fungsi `MinMax` akan menambahkan nilai **1** ke dalam M . Anda akan menerima nilai penuh untuk subsoal ini apabila $M \leq \frac{N+1}{2}$ untuk semua kasus uji.

Subsoal 2 (70 poin): Dimisalkan k adalah banyaknya bilangan-bilangan pada masukan yang lebih besar atau sama dengan s dan lebih kecil atau sama dengan t pada sebuah pemanggilan fungsi `MinMax`. Setiap pemanggilan fungsi `MinMax` akan menambahkan nilai $k + 1$ ke dalam M . Nilai akhir akan dihitung dengan aturan berikut: Nilai akhir untuk subsoal ini merupakan nilai minimum yang Anda peroleh diantara semua kasus uji yang diberikan. Untuk sebuah kasus uji, nilai yang diperoleh adalah 70

apabila $M \leq 3N$ dan nilai yang diperoleh adalah $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$, jika tidak.

Percobaan

Sample grader yang dapat diunduh dari sistem penilaian akan membaca data dari standar masukan. Baris pertama dari masukan berisi dua buah bilangan bulat, nomor subsoal T , dan N . Baris berikutnya berisi N buah bilangan bulat yang terurut menaik. *Sample grader* akan menulis ke standar keluaran sebuah nilai yang dikembalikan oleh fungsi `findGap` di baris pertama dan nilai dari M sesuai dengan subsoal dimana kasus uji tersebut berada.

Masukan berikut mendeskripsikan contoh di atas:

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