



Gap

N 개의 음이 아닌 정수 a_1, a_2, \dots, a_N 가 있어서 다음 부등식을 만족한다고 한다:

$0 \leq a_1 < a_2 < \dots < a_N \leq 10^{18}$. 지학이는 i 가 1부터 $N - 1$ 까지 변할 때 $a_{i+1} - a_i$ 의 최대값을 알고 싶다. 입력 자연수들은 직접 지학이의 프로그램에 주어지는 것이 아니고, 특정한 함수를 이용하여 접근할 수 있다. 사용하는 프로그래밍 언어에 대한 Implementation 부분을 보면 상세한 내용을 알 수 있다.

Task

지학이를 도와 i 가 1부터 $N - 1$ 까지 변할 때 $a_{i+1} - a_i$ 의 최대값을 구하는 함수를 구현하라.

Implementation for C and C++

구현할 함수는 `findGap(T, N)`이다. 이 함수는 다음의 인자를 받아서 `long long` type의 자연수 값을 리턴한다:

- T — subtask 번호 (1 혹은 2)
- N — 입력 자연수의 개수

구현할 함수 `findGap` 안에서 함수 `MinMax(s, t, &mn, &mx)`를 호출할 수 있다. 호출에서 인자 s 와 t 는 `long long` type의 자연수이고 인자 $\&mn$ 와 $\&mx$ 은 `long long` type의 자연수 변수에 대한 포인터이다. 즉, mn 과 mx 은 `long long` type의 자연수 변수이다. `MinMax(s, t, &mn, &mx)`가 리턴한 직후 mn 의 값은 s 보다 크거나 같은 a_i 들 중 최소값이 되고, mx 의 값은 t 보다 작거나 같은 a_j 의 값들 중 최대값이 된다. 만약 s 이상이고 t 이하인 입력 자연수가 존재하지 않는 경우라면 mn 과 mx 의 값은 모두 -1 이 된다. `MinMax`가 호출될 때 s 의 값은 t 의 값보다 작거나 같아야 한다. 이 조건이 만족되지 않는 경우 프로그램은 0 이 아닌 리턴 값으로 종료된다.

Implementation for Pascal

구현할 함수는 `findGap(T, N)`이다. 이 함수는 다음의 인자를 받아서 `Int64` type의 자연수 값을 리턴한다:

- T — subtask 번호 (1 or 2) (Integer type)
- N — 입력 자연수의 개수 (LongInt type)

구현할 함수 `findGap` 안에서 함수 `MinMax(s, t, mn, mx)`를 호출할 수 있다. 호출에서 인자 s 와 t 는 `Int64` type의 자연수이고 인자 mn 와 mx 은 `Int64` type의 자연수 변수를 **call by reference**로 전달한 것이다. 즉, mn 과 mx 은 `long long` type의 자연수 변수이다. `MinMax(s, t, mn, mx)`가 리턴한 직후 mn 의 값은 s 보다 크거나 같은 a_i 들 중 최소값이 되고, mx 의 값은 t 보다 작거나 같은 a_j 의 값들 중 최대값이 된다. 만약 s 이상이고 t 이하인 입력 자연수가 존재하지 않는 경우라면 mn 과 mx 의 값은 모두 -1 이 된다. `MinMax`가 호출될 때 s 의 값은 t 의 값보다 작거나 같아야 한다. 이 조건이 만족되지 않는 경우 프로그램은 종료된다.

Implementation for all

일반적인 제한조건(시간, 메모리, 실행시간 오류 없음, 등)이외에 다음의 조건을 만족해야 테스트 케이스를 푼 것으로 간주된다:

- findGap 함수는 정확한 답을 리턴해야 하고,
- MinMax를 호출하는 데 따르는 비용 M 은 허용된 제한을 넘으면 한된다. (Scoring 부분을 보라.)

Example for C, C++

입력이 다음과 같다고 하자: $N = 4$ 이고 $a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$.

이 때 답은 3인데, findGap 함수에서 MinMax함수를 다음과 같이 호출했다면 답을 알수 있다:

- MinMax(1, 2, &mn, &mx) 가 호출되어 mn과 mx는 모두 값이 2이다.
- MinMax(3, 7, &mn, &mx) 가 호출되어 mn의 값은 3이고 mx의 값은 6이다.
- MinMax(8, 9, &mn, &mx) 가 호출되어 mn과 mx는 모두 값이 8이다.

Example for Pascal

입력이 다음과 같다고 하자: $N = 4$ 이고 $a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 8$.

이 때 답은 3인데, findGap 함수에서 MinMax함수를 다음과 같이 호출했다면 답을 알수 있다:

- MinMax(1, 2, mn, mx) 가 호출되어 mn과 mx는 모두 값이 2이다.
- MinMax(3, 7, mn, mx) 가 호출되어 mn의 값은 3이고 mx의 값은 6이다.
- MinMax(8, 9, mn, mx) 가 호출되어 mn과 mx는 모두 값이 8이다.

Scoring

모든 subtask에서 $2 \leq N \leq 100,000$ 이 성립한다.

Subtask 1 (30 점): MinMax를 한번 호출할 때 마다 M 에 1이 더해진다. 모든 테스트 케이스에 대해 $M \leq \frac{N+1}{2}$ 인 경우 30점을 받는다.

Subtask 2 (70 points): MinMax가 호출될 때 s 이상이고 t 이하인 입력 자연수의 개수를 k 라고 하자. MinMax를 호출할 때 마다 $k + 1$ 이 M 에 더해진다. 최종 점수가 계산되는 규칙은 다음과 같다: 이 subtask의 최종 점수는 프로그램이 테스트 케이스들에서 받은 점수의 최소값이다. 한 테스트 케이스에 대해서, $M \leq 3N$ 라면 70점이 주어지고, 그렇지 않다면 $\frac{60}{\sqrt{\frac{M}{N}+1}-1}$ 로 계산된 값이 주어진다.

다.

Experimentation

채점 시스템에서 다운로드 받을 수 있는 sample grader는 표준 입력에서 데이터를 읽어들인다. 입력의 첫 줄에는 두개의 자연수가 있어야 하는데, 이들은 subtask 번호 T 와 N 이다. 다음 줄에는 N 개의 자연수가 증가하는 순서로 있어야 한다. Sample grader는 표준 출력으로 findGap의 리턴 값을 첫 줄에 출력하고 입력의 subtask에 해당하는 M 값을 두번째 줄에 출력한다.

다음 입력은 위의 예제에 해당하는 것이다.

2 4
2 3 6 8