P4. Step Function

문제정의

두 개의 Step function f, g와 구간 [p,q]가 주어지고, 구간 내 모든 정수에 대하여 각 각의 함수값을 비교하는 문제

유의할 점

f, g의 Domain과 Range는 Real number로 명시돼 있다 - > [long double type과 fmodl 함수를 사용한다]

f, g의 함수값은 negative 값을 가질 수 있고, 이전 step보다 작은 값을 가질 수 있다

입력이 크기 때문에 큰 byte의 자료형에서도 overflow가 일어날 수 있다 -> [연산 중간에 modular 연산을 섞는다]

구현 전략

1. Merge: 임의의 입력 a에 대해 항상 max(f(a), g(a)) 값을 갖는 새로운 step function h를 구한다
2. Scan: 구간 [p,q]에 해당하는 h의 함수값을 찾아 더한다

Merge

1. f, g 배열 원소의 모든 x 값은 h 배열 원소의 x 값이 된다 (자료구조 부분의 표현방식을 사용한다)
2. x 값은 Merge sort의 부분 procedure와 유사하게, 두 배열의 x 값 중 더 작은 값을 기준으로 합친다
3. 이 때 각 x에 해당하는 y 값은 x가 속한 함수의 함수 값과, 다른 함수의 함수 값을 비교하여 큰 것으로 한다

Scan

h의 x(i)값을 left 다음 x(i+1)값을 right라 할 때 (1≤ i ≤ size(h) - 2), left < right 이고 p ≤ q 이다.

부분 구간 [left, right)와 구간 [p,q]의 관계에 따라, 해당 부분 구간에서 구간 [p,q]에 속하는 값의 수를 구할 수 있다.

이를 cnt라고 하면, cnt * h(x(i))를 모든 구간에 구해 더하고 modular 연산을 한 것이 정답이 된다.

이 때 (cnt mod 10007 * h(x(i)) mod 10007) mod 10007 로 overflow를 막을 수 있다.

자료구조

Step function의 각 구간을 시작하는 x와

각 구간의 함수값 y인 (x,y) 를 원소로 갖는 배열로

함수 f, g, h를 표현할 수 있다

size(h) ≤ size(f) + size(g)

시간 복잡도

배열 h의 크기를 n이라 할 때, n은 입력의 크기보다 같거나 작다

$O(n)$ - h를 만들 때

$O(n)$ - 구간에 속하는 모든 함수값의 합을 구할 때

p ≤ q < left < right : cnt = 0

p ≤ left ≤ q ≤ right : cnt = q - left + 1 || q - left (right==q)

p ≤ left < right < q : cnt = right - left

left < p < right ≤ q : cnt = right - p

left < p ≤ q ≤ right : cnt = q - p + 1 || q - p (right==q)

left < right ≤ p ≤ q : cnt = 0

21500800 홍순규