

## 문제정의

f함수와 g함수를 합하였을 때 가장 큰 합이 나오는 새로운 함수를 생성하여 주어진 p,q안의 최대값을 찾는 문제

## 문제해결

### 입력

시작점과 함수값을 함께 저장하는 클래스를 이용하여 f 함수를 정하는 f 배열과 g함수를 저장하는 g 배열에 각 함수의 시작점과 함수값을 각각 입력 받는다. 정답을 저장하는 변수 res는 0으로 초기화한다

### 전처리

f함수와 g함수의 시작점들을 오름차순으로 하여 두 함수를 합해 새로운 배열을 생성한다(이하 fg배열). 문제에서 구하는 바는 p와 q사이의 최대값이다. 따라서 시작점이 p미만인 원소와 q초과인 원소는 fg배열에서 삭제한다

### 최대값 구하기

현재 지점 x가 p부터 시작하여 fg배열의 끝에 도달할 때까지 함수값이 변하는 지점을 y를 fg배열에서 받아온다. y의 값을 받아올때마다 x에서 y까지의 함수값의 합을 구한다. 이때 문제에서 요구하는 것은 10007을 나눈 나머지 값이므로 <수식1>을 적용한다.

<수식1>

x에서 y사이의 최대값의 10007로 나눈 나머지를 max(x,y)라고 할 때,

$$\max(x,y) = ((y-x)\%10007) * (\text{value}\%10007), \text{value} = \max(f(x), g(x))$$

x와 y의 값을 y와 새로운 변환점 y'로 업데이트 하면서 max(x,y)를 res에 더해나간다. 더하고 나서 res값에 다시 나머지연산을 취한다. y가 fg배열의 끝까지 도달하여도 q에 도달하지 못한 경우에는 (q+1 -x)%10007 \* (value%10007)을 더해준 뒤, 다시 10007로 나머지 연산을 시행한다.

### 타당성

$$\max(x,y) = \frac{a}{b} * \frac{b}{c}$$

$$\overline{\max(x,y)}_n = \overline{a}_n * \overline{b}_n, n \text{ is modulus in this problem } n = 10007$$

$$\overline{res}_n = \overline{\max_1(x,y)}_n + \overline{\max_2(x,y)}_n + \dots$$

### 시간복잡도

Kf와 Gf의 합을 n이라 할 때 최악의 경우 모든 배열의 원소를 한번씩 접근하므로 O(n)이 된다.

### 공간복잡도

Kf+ Gf의 최대값인 2000000으로 배열의 크기를 잡았으므로 공간복잡도는 O(1)이다.

### 응용

f와 g함수를 다른 배열에 저장하는 것이 아니라 하나의 배열로 합쳤을 때 논리를 짜기 훨씬 수월하였다. 또한 전처리를 넣었을 때 실행속도가 빨라지는 경우가 있었다. 알고리즘을 구상하고 나서 알고리즘을 가장 빨리 돌아갈 수 있는 데이터처리에 대해서도 고민할 수 있었다.