

지각 (tardy)

처음반 학생인 은수는 늦잠을 잔 나머지 강의 시간에 늦고 말았다. 은수는 퇴교를 면하기 위해 코치들이 눈치채지 못한 사이 강의실에 몰래 들어가려고 한다.

하지만 코치들은 그렇게 쉽게 지각생을 들여보내지 않는다. 코치들은 지각생들이 몰래 강의에 참여할 수 없도록, 오전 8시 50분에 참여 링크를 바꿔 놓았다.

구체적으로, 지각생을 막기 위해서 코치들은 $N + 1$ 개의 주소를 사용한다. $N + 1$ 번째 주소는 강의실에 참여할 수 있는 유일한 주소이며, 1번째부터 N 번째까지의 주소는 지각생이 몰래 강의실에 들어가는 것을 막기 위하여 사용되는 주소로, $N + 1$ 개의 주소 중 하나로 가는 링크를 알려주는 주소이다.

또한, 은수를 포함한 처음반 학생들은 정수로 표현되는 암호화된 아이디 하나를 가지고 있다. 은수가 코치들에게 받은 주소인 S 번 주소에 접속할 때의 은수의 아이디 값은 id_0 이다.

학생이 i 번째($1 \leq i \leq N$) 주소에 접속하였을 때는, 학생의 아이디가 특정한 값 A_i 와 같은지에 따라서 서로 다른 링크 두 개 중 하나를 알려준다.

만약 학생의 아이디가 A_i 라면, 링크를 클릭할 때 학생의 아이디 값에 B_i 가 더해진다. 또한, 이 링크를 통해서 C_i 번째 주소에 접속할 수 있다.

만약 학생의 아이디가 A_i 가 아니라면, 링크를 클릭할 때 학생의 아이디 값에 D_i 가 더해진다. 또한, 이 링크를 통해서 E_i 번째 주소에 접속할 수 있다.

은수는 정확히 9시에 S 번 주소에 접속하였고, 알려주는 링크를 클릭하는 데 걸리는 시간은 1초이다. 은수는 강의실에 최대한 빨리 들어가고 싶기 때문에 강의실에 들어갈 때까지 쉬지 않고 다음 주소에 접속할 것이다. 주소는 복잡하고 길기 때문에, 은수는 현재 주소에서 알려주는 링크만을 클릭할 것이다.

물론, 코치들은 이미 은수의 지각을 알고 있다. 은수의 지각을 이미 눈치챈 코치들은 은수가 강의 시작으로부터 몇 초나 늦게 강의실에 들어올지를 가지고 학생들과 저녁 시간 내기를 하는 중이다. 정확히는, 은수가 늦은 시간을 초 단위로 나타낸 후 $10^9 + 7$ 로 나눈 나머지를 가지고 내기를 하고 있다.

은수가 강의실에 들어오는 시각은 은수가 $N + 1$ 번 링크를 클릭하는 시각과 정확히 같으며, 강의는 은수가 S 번 주소에 접속한 때인 9시 정각에 시작하였다.

당신은 이 내기를 이기기 위해서, 은수가 늦을 시간을 $10^9 + 7$ 로 나눈 나머지를 계산하고자 한다. 만약 은수가 아무리 노력해도 강의실에 들어올 수 없다면, 내기를 거절하기 위해 -1 을 제시하여야 한다.

구현 세부 사항

여러분은 아래 함수를 구현해야 한다.

```
int tardy(int S, int64 id, int64[] A, int64[] B, int[] C, int64[] D, int[] E)
```

- 정수 S 는 S 를 나타낸다.
- 정수 id 는 id_0 를 나타낸다.
- 정수 배열 A 는 길이 N 의 수열로, $A[0], A[1], \dots, A[N-1]$ 에 A_1, A_2, \dots, A_N 이 저장되어 있다.
- 정수 배열 B 는 길이 N 의 수열로, $B[0], B[1], \dots, B[N-1]$ 에 B_1, B_2, \dots, B_N 이 저장되어 있다.
- 정수 배열 C 는 길이 N 의 수열로, $C[0], C[1], \dots, C[N-1]$ 에 C_1, C_2, \dots, C_N 이 저장되어 있다.
- 정수 배열 D 는 길이 N 의 수열로, $D[0], D[1], \dots, D[N-1]$ 에 D_1, D_2, \dots, D_N 이 저장되어 있다.
- 정수 배열 E 는 길이 N 의 수열로, $E[0], E[1], \dots, E[N-1]$ 에 E_1, E_2, \dots, E_N 이 저장되어 있다.
- 이 함수는 은수가 강의 시작으로부터 몇 초나 늦을지를 $10^9 + 7$ 로 나눈 값을 반환해야 한다. 만약 은수가 강의실에 영원히 들어오지 못한다면 -1 을 반환해야 한다.

제약 조건

- $1 \leq N \leq 100\,000$
- $1 \leq S \leq N$
- $-10^{13} \leq id_0 \leq 10^{13}$
- $-10^{13} \leq A_i \leq 10^{13}$ ($1 \leq i \leq N$)
- $-10^{13} \leq B_i \leq 10^{13}$ ($1 \leq i \leq N$)
- $1 \leq C_i \leq N + 1$ ($1 \leq i \leq N$)
- $-10^{13} \leq D_i \leq 10^{13}$ ($1 \leq i \leq N$)
- $1 \leq E_i \leq N + 1$ ($1 \leq i \leq N$)

부분문제

1. (14점) $C_i = N + 1$ ($1 \leq i \leq N$)
2. (19점) $N \leq 5\,000$
3. (21점) $E_i > i$ ($1 \leq i \leq N$)
4. (25점) $E_i = i + 1$ ($1 \leq i \leq N - 1$), $E_N = 1$
5. (21점) 추가 제한 조건 없음.

예제

- $N = 3, S = 1, id_0 = 4, A = [2, -2, 1], B = [-2, 0, 3], C = [4, 4, 4], D = [2, 3, -6], E = [2, 3, 1]$. 그레이더는 다음의 함수를 호출한다.

$\text{tardy}(3, 1, 4, [2, -2, 1], [-2, 0, 3], [4, 4, 4], [2, 3, -6], [2, 3, 1]) = 7$

이 예제는 1, 2, 4, 5번 부분문제의 조건을 만족한다.

- $N = 5, S = 2, id_0 = 1, A = [1, 1, -1], B = [-3, 0, 2], C = [2, 1, 1], D = [1, 1, 1], E = [4, 3, 4]$. 그레이더는 다음의 함수를 호출한다.

$\text{tardy}(3, 2, 1, [1, 1, -1], [-3, 0, -2], [2, 1, 1], [1, 1, 1], [4, 3, 4]) = -1$

이 예제는 2, 3, 5번 부분문제의 조건을 만족한다.

- $N = 5, S = 3, id_0 = -3, A = [1, -2, 9, 29, 3], B = [2, 4, -6, 3, -5], C = [5, 3, 2, 6, 1], D = [2, -1, 4, -1, 2], E = [4, 1, 1, 3, 6]$. 그레이더는 다음의 함수를 호출한다.

$\text{tardy}(5, 3, -3, [1, -2, 9, 29, 3], [2, 4, -6, 3, -5], [5, 3, 2, 6, 1], [2, -1, 4, -1, 2], [4, 1, 1, 3, 6]) = 30$

이 예제는 2, 5번 부분문제의 조건을 만족한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: $N \ S \ id_0$
- Line $1 + i$: $A_i \ B_i \ C_i \ D_i \ E_i \ (1 \leq i \leq N)$