

날다람쥐 2

2차원 평면 위에 날다람쥐와 N 개의 기둥이 있다.

2차원 평면 위의 한 점은 (x, y) 꼴로 표현할 수 있다. 이때, x 는 수평 위치, y 는 높이를 의미한다. 수평 위치 x 가 증가하는 방향은 오른쪽 방향이며, 높이 y 가 증가하는 방향은 위쪽 방향이다.

기둥은 0번부터 $N - 1$ 번까지의 번호가 순서대로 붙어 있다. i ($0 \leq i \leq N - 1$)번 기둥의 바닥은 점 $(i, 0)$ 에 위치하고, 높이는 무한하다. 따라서 i 번 기둥은 점 $(i, 0)$ 에서 시작해서 위로 뻗은 반직선이다. 각 기둥은 빨간색 또는 파란색이다. $B[i] = 0$ 이면 i 번 기둥은 빨간색이고, $B[i] = 1$ 이면 i 번 기둥은 파란색이다.

처음에 날다람쥐는 점 $(0, 0)$ 에 있다. 날다람쥐는 점 (N, H) 에 도착하려고 한다. 이를 위해 날다람쥐는 다음과 같은 방식으로 이동한다.

날다람쥐는 기둥이 없는 곳에서는 자신의 현재 높이를 유지하며 오른쪽으로 날아간다. 날다람쥐는 매우 빠르기 때문에 이 때 소요되는 시간은 0으로 간주한다.

날다람쥐는 기둥이 있는 곳에서는 날다람쥐가 높이를 1만큼 증가시키거나 아무것도 하지 않을 수 있다. 구체적으로, 날다람쥐는 i ($0 \leq i \leq N - 1$)번 기둥이 있는 곳에서 다음과 같은 행동 중 하나를 해야 한다.

- 기둥을 지나쳐 간다. 날다람쥐의 높이가 그대로 유지되며, 날다람쥐는 계속 오른쪽으로 날아간다. 소요되는 시간은 0이다.
- 기둥을 기어올라간다. 이 행동은 기둥이 빨간색일 때만 가능하다 ($B[i] = 0$). 날다람쥐의 높이는 기둥에서 1만큼 증가하며, 그 후 다시 오른쪽으로 날아간다. 소요되는 시간은 $A[i]$ 이다.
- 기둥에서 점프한다. 이 행동은 기둥이 파란색일 때만 가능하다 ($B[i] = 1$). 날다람쥐의 높이는 기둥에서 1만큼 증가하며, 그 후 다시 오른쪽으로 날아간다. 소요되는 시간은 $A[i]$ 이다.

추가로, 날다람쥐가 수평 위치 $i + 0.5$ ($0 \leq i \leq N - 1$)인 지점을 지나갈 때, 날다람쥐의 높이는 $L[i]$ 이상, $R[i]$ 이하여야 한다. 날다람쥐가 수평 위치 N 인 지점에 도착할 때, 날다람쥐의 높이는 정확히 H 여야 한다.

위 조건을 모두 만족하며 (N, H) 에 도달하는 방법 중 날다람쥐가 점프한 횟수가 정확히 k 번일 때 소요되는 총 시간의 최솟값을 $T[k]$ 라고 정의하자. 만약 그러한 방법이 존재하지 않는다면 $T[k] = -1$ 로 정의한다.

$T[0], T[1], \dots, T[H]$ 를 구하여라.

함수 목록 및 정의

다음 함수를 구현해야 한다.

```
vector<long long> fly(int H, vector<int> A, vector<int> B, vector<int> L,  
vector<int> R)
```

- H : 날다람쥐의 최종 높이
- A, B, L, R : 크기가 N 인 정수 배열
- B : 기둥의 색을 나타내는 배열. $B[i] = 0$ 이면 i 번 기둥의 색은 빨간색, $B[i] = 1$ 이면 i 번 기둥의 색은 파란색이다.
- 이 함수는 크기가 $H + 1$ 인 배열 T 를 반환해야 한다.
- 이 함수는 단 한 번만 호출된다.

제약 조건

- $1 \leq N \leq 200\,000$
- $0 \leq H \leq N$
- $0 \leq A[i] \leq 10^9$ ($0 \leq i \leq N - 1$)
- $0 \leq B[i] \leq 1$ ($0 \leq i \leq N - 1$)
- $0 \leq L[i] \leq R[i] \leq N$ ($0 \leq i \leq N - 1$)

부분문제

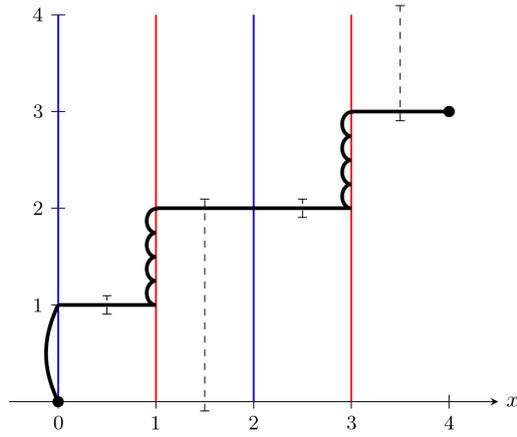
번호	배점	제한
1	3	$N \leq 300$
2	4	$A[i] = B[i] = 0$ ($0 \leq i \leq N - 1$)
3	25	$B[i] = 0$ ($0 \leq i \leq N - 1$)
4	20	$N \leq 65\,000, A[i] \leq 5$ ($0 \leq i \leq N - 1$)
5	29	$N \leq 65\,000$
6	19	추가적인 제약조건이 없다.

예제

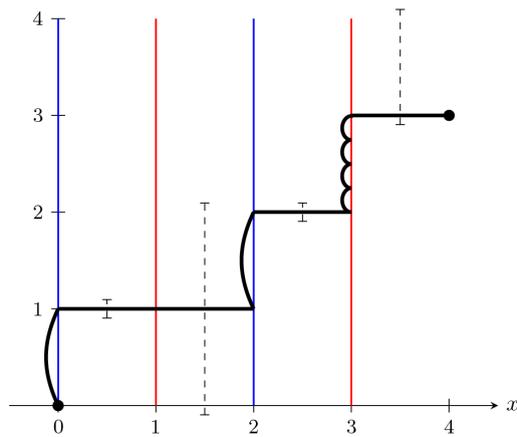
예제 1

다음 호출을 생각해 보자.

```
fly(3, [8, 8, 2, 4],  
     [1, 0, 1, 0],  
     [1, 0, 2, 3],  
     [1, 2, 2, 4])
```



0번 기둥에서 점프하고 1번, 3번 기둥을 기어올라가면 조건을 만족한다. 이 경우, 점프 횟수는 1번이고 소요된 시간은 20초이다.



0번, 2번 기둥에서 점프하고 3번 기둥을 기어올라가면 조건을 만족한다. 이 경우, 점프 횟수는 2번이고 소요된 시간은 14초이다.

가능한 다른 방법은 존재하지 않는다. 따라서, 함수는 $[-1, 20, 14, -1]$ 을 반환해야 한다.

예제 2

다음 호출을 생각해 보자.

```
fly(1, [1000000000],
     [0],
     [1],
     [1])
```

함수는 $[1000000000, -1]$ 를 반환해야 한다.

예제 3

다음 호출을 생각해 보자.

```
fly(3, [4, 7, 0, 3, 8, 4, 5],
      [0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 1, 2],
      [5, 1, 2, 5, 5, 6, 3])
```

함수는 $[7, -1, -1, -1]$ 를 반환해야 한다.

예제 4

다음 호출을 생각해 보자.

```
fly(7, [3, 3, 4, 1, 3, 2, 0, 1, 4, 3, 4, 0, 0, 1, 0, 4, 4, 5, 5, 0],
      [1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1],
      [0, 0, 1, 1, 2, 1, 2, 2, 1, 1, 0, 1, 1, 3, 2, 2, 1, 6, 4, 4],
      [3, 2, 3, 3, 6, 2, 2, 4, 3, 4, 4, 5, 3, 6, 6, 5, 7, 8, 8, 9])
```

함수는 $[-1, 16, 11, 10, 9, 10, 12, 15]$ 를 반환해야 한다.

샘플 그레이더

샘플 그레이더의 입력 형식은 다음과 같다.

- line 1: $N H$
- line 2: $A[0] A[1] \dots A[N - 1]$
- line 3: $B[0] B[1] \dots B[N - 1]$
- line 4: $L[0] L[1] \dots L[N - 1]$
- line 5: $R[0] R[1] \dots R[N - 1]$

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line 1: $T[0] T[1] \dots T[H]$