

총 쏘기

두 명이 같이 하는 인터넷 슈팅 게임이 있다. 이 게임은 폐허가 된 도시에서 빌딩들을 부수는 게임이다. 게임에서 바닥인 수평선 위에 N 개의 빌딩이 왼쪽에서 오른쪽으로 서 있다. 빌딩은 왼쪽에서 오른쪽으로 순서대로 1부터 N 의 정수로 나타낸다. 각 빌딩의 바닥으로부터의 높이는 수열 A_i ($1 \leq i \leq N$)로 나타내고, 1부터 N 까지의 서로 다른 정수로 주어진다.

두 명의 플레이어는 모든 빌딩보다 왼쪽의 같은 위치에 있다. 시간 i ($i \geq 1$)에 두 명의 플레이어는 동시에 각 한발씩 총을 발사하고, 총알은 발사한 위치에서 수평으로 오른쪽으로 날아간다. 두 총알의 속도는 동일하다. 플레이어는 총알의 발사 높이를 바닥으로부터의 거리 H 로 결정한다. H 는 1이상 $N + 1$ 이하의 정수이다. 두 플레이어는 동일한 발사 높이를 선택할 수 있다.

플레이어의 총알 발사 높이가 H 인 경우, $A_i \geq H$ 를 만족하는 파괴되지 않은 가장 왼쪽의 빌딩이 이 총알로 파괴된다. 이 조건을 만족하는 빌딩이 없다면, 아무 일도 일어나지 않는다. 만약 두 플레이어가 발사한 총알에 대해 이 조건을 만족하는 빌딩이 동일하다면, (두 총알의 속도는 동일하기 때문에) 이 하나의 빌딩만 파괴된다. 특별히, 두 플레이어의 발사 높이가 같다면, 항상 하나의 빌딩만 파괴된다. 예를 들어, $A_1 = 2, A_2 = 1$ 이고, 처음에 두 플레이어가 모두 $H = 1$ 을 발사 높이로 결정하였다면, 이 두 총알로 빌딩 1만 파괴된다.

문제는 N 개 빌딩들의 높이가 입력으로 주어질 때, 모든 빌딩을 파괴할 수 있는 최소 시간과 각 시간에 두 플레이어의 총알 발사 높이를 찾는 것이다.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
vector< pair<int, int> > min_shooting_buildings(vector<int> A)
```

- 이 함수는 단 한 번만 호출된다.
- 인자로 주어지는 배열 A 의 크기는 N 이다. 배열의 각 원소 $A[i]$ 는 빌딩 $i + 1$ 의 높이 A_{i+1} 을 나타낸다 ($0 \leq i \leq N - 1$).
- 이 함수는 두 플레이어가 모든 빌딩을 파괴하기 위한 최소 시간 S 에 대해서 크기 S 인 배열 M 을 반환한다. 배열 M 의 각 원소 (a, b) 에서 a 와 b 는 각각 첫 번째와 두 번째 플레이어의 총알 발사 높이를 나타낸다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \leq N \leq 100\,000$
- $1 \leq A_i \leq N$ ($1 \leq i \leq N$)
- A_i ($1 \leq i \leq N$)는 모두 서로 다르다.

부분문제

1. (17점)
 - $1 \leq i < j < k \leq N$ 이고 $A_i < A_j < A_k$ 를 만족하는 (i, j, k) 가 존재하지 않는다.
2. (12점)
 - $1 \leq i < j < k \leq N$ 이고 $A_i > A_j > A_k$ 를 만족하는 (i, j, k) 가 존재하지 않는다.
3. (9점)
 - $N \leq 4$
4. (12점)
 - $N \leq 16$
5. (31점)
 - $N \leq 500$
6. (29점)
 - $N \leq 7\,500$
7. (40점)
 - 추가적인 제약 조건이 없다.

채점 기준

`min_shooting_buildings` 함수가 반환한 배열의 크기가 두 플레이어가 모든 빌딩을 파괴하는 최소 시간 S 와 같고, 반환한 배열의 원소들과 같이 두 플레이어가 총알을 발사할 때 모든 빌딩들이 파괴되면 해당 데이터를 맞춘 것으로 채점한다.

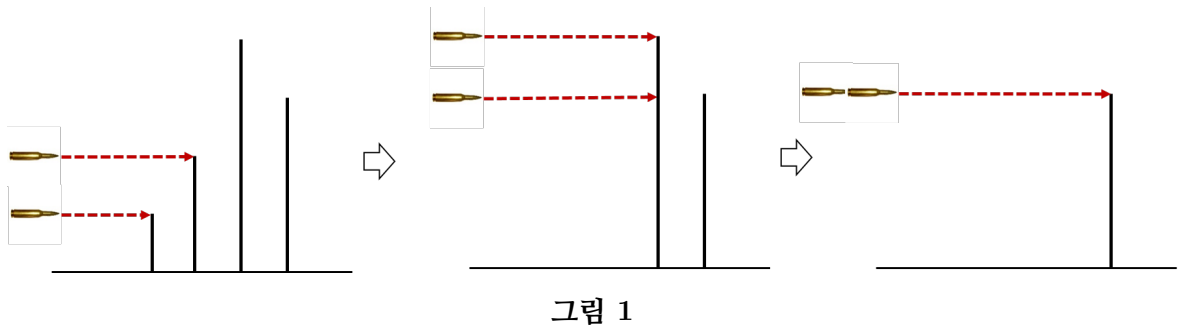
예제

- $N = 4$, 인자 $A = [1, 2, 4, 3]$ 인 경우를 생각해 보자.

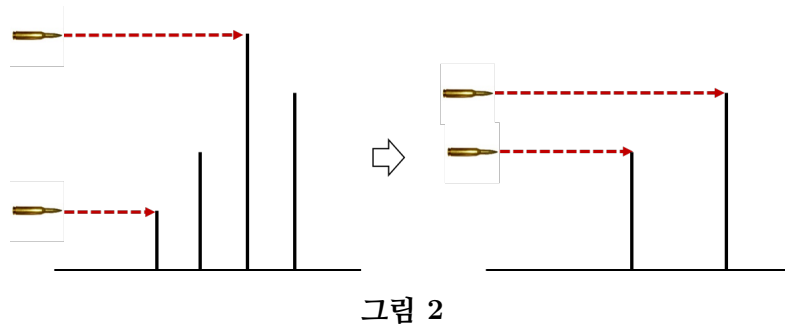
그레이더는 다음 함수를 호출한다.

```
min_shooting_buildings([1, 2, 4, 3])
```

아래 **그림 1**과 같이 두 플레이어의 총알 발사 위치를 (1,2), (3,4), (3,3)로 결정하면 시간 3만에 빌딩을 모두 파괴한다.



아래 **그림 2**와 같이 두 플레이어의 총알 발사 위치를 (1,4), (2,3)로 결정하면 시간 2만에 빌딩을 모두 파괴한다.



`min_shooting_buildings` 함수는 종료하면서 크기 2인 배열을 리턴해야 하고 배열의 가능한 내용 중 하나는 [(1, 4), (2, 3)]이다.

- $N = 8$, 인자 $A = [4, 3, 8, 2, 1, 7, 6, 5]$ 인 경우를 생각해 보자.

`min_shooting_buildings` 함수는 종료하면서 크기 4인 배열을 리턴해야 하고 배열의 가능한 내용 중 하나는 [(4, 8), (3, 7), (2, 6), (1, 5)]이다.

- $N = 8$, 인자 $A = [5, 6, 7, 1, 2, 8, 3, 4]$ 인 경우를 생각해 보자.

`min_shooting_buildings` 함수는 종료하면서 크기 4인 수열을 리턴해야 하고 배열의 가능한 내용 중 하나는 [(5, 6), (7, 8), (1, 2), (3, 4)]이다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N
- Line 2: $A[0] A[1] \cdots A[N - 1]$

Sample grader는 다음을 출력한다.

- Line i ($1 \leq i \leq S$): 함수 `min_shooting_buildings`가 반환한 배열의 i 번째 원소

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.