

# 정렬하기

앨리스와 밥이 게임을 하고 있다. 앨리스는 0번부터  $N - 1$ 번까지 번호가 붙은  $N$ 개의 물건을 가지고 있다. 각 물건  $i$ 의 가치는 음이 아닌 정수  $A[i]$ 이다.

앨리스는 모든 물건의 가치를 알고 있지만, 밥은 물건의 개수  $N$ 과 각 물건의 가치가 음이 아닌 정수라는 사실만 알고 있다. 밥의 목표는 물건들을 가치가 감소하지 않는 순서로 정렬하는 것이다. 즉, 밥은 다음을 만족하는 길이가  $N$ 인 정수 배열  $P$ 를 알아내야 한다.

- $P$ 의 각 원소는 서로 다르다.
- $P$ 의 모든 원소는 0 이상  $N - 1$  이하이다.
- 모든  $0 \leq i \leq N - 2$ 을 만족하는 정수  $i$ 에 대하여  $A[P[i]] \leq A[P[i + 1]]$

이를 위해 밥은 앨리스에게 최대 10 000번의 질문을 할 수 있다. 각 질문은 다음과 같이 진행된다.

1. 밥은 서로 다른 두 물건을 잇는 실  $N - 1$ 개를 설치한다. 이때 임의의 두 물건 사이를 실들을 따라 이동할 수 있도록 전체가 하나로 연결되어야 한다.
2. 앨리스는 다음 규칙에 따라 0개 이상의 물건을 선택한다.
  - 실로 직접 연결된 두 물건을 동시에 고를 수 없다.
  - 위 조건을 만족하면서 고른 물건들의 가치 합이 가능한 한 최대가 되어야 한다.

만약 조건을 만족하는 물건 조합이 여러 가지라면, 앨리스는 그중 하나를 임의로 선택해 알려준다.

3. 앨리스는 고른 물건들을 밥에게 알려주고, 실을 모두 제거한다.

여러분은 밥을 도와 최대한 적은 수의 질문을 통해 게임에서 승리해야 한다.

## 함수 목록 및 정의

다음 함수를 구현해야 한다.

```
vector<int> sorting(int N)
```

- $N$ : 게임에 사용되는 물건의 개수
- 이 함수는 가치가 감소하지 않는 순서대로 나열된 물건 번호들의 배열  $P$ 를 반환해야 한다. 만약 이러한 조건을 만족하는 배열  $P$ 가 유일하지 않을 경우, 조건을 만족하는 배열 중 어느 것을 반환해도 된다.
- 이 함수는 단 한 번만 호출된다.

이 함수는 다음 함수를 호출할 수 있다.

```
vector<int> ask_question(vector<array<int, 2>> threads)
```

- 앨리스와 밥이 한 번의 질문 과정을 진행함을 의미한다.
- `threads`: 실로 직접 연결하는 물건들의 쌍을 나타내는 크기  $N - 1$ 의 배열. `threads`의 각 원소  $[a, b]$ 에 대해  $a$ 번 물건과  $b$ 번 물건을 연결하는 실을 설치함을 의미한다.
- `threads`에 명시된 대로 물건들을 연결했을 때 임의의 두 물건 사이를 실들을 따라 이동할 수 있어야 한다.
- 이 함수는 길이가  $N$ 인 정수 배열  $C$ 를 반환한다. 각 물건  $i$ 에 대해 앨리스가 이번 질문에서  $i$ 번 물건을 고른다면  $C[i] = 1$ , 그렇지 않다면  $C[i] = 0$ 이다. ( $0 \leq i \leq N - 1$ )
  - 이번 질문에서 앨리스가 조건을 만족하는 물건 조합이 유일하지 않을 경우, 그레이더는 조건을 만족하는 배열  $C$  중 하나를 선택하여 반환한다. 한 테스트케이스에서 같은 `threads` 배열을 인자로 한 여러 번의 `ask_question` 호출이 서로 다른 배열을 반환할 수 있음에 유의하라.
- 이 함수는 한 테스트케이스에서 최대 10 000번 호출할 수 있다.

## 제약 조건

- $5 \leq N \leq 1000$
- $A[i]$ 는 음이 아닌 정수이다.  $A[i]$ 의 상한은 주어지지 않음에 유의하라. ( $0 \leq i \leq N - 1$ )
- `ask_question` 함수는 한 테스트 케이스에서 최대 10 000번 호출될 수 있다.
- 이 문제에서 그레이더는 적응적(adaptive)이다. 이것은 배열  $A$ 가 고정되어 있지 않고 `ask_question` 함수 호출에 따라 변할 수 있음을 의미한다. 그레이더가 매번 답변을 할 때마다 이전에 있었던 모든 `ask_question` 호출의 결과와 일치하는 배열  $A$ 가 적어도 하나 이상 존재하는 것은 보장된다.
- 모든 테스트케이스에 대하여 그레이더는 `ask_question` 호출과 상관없이 2초 이내의 시간을 소모하며, 16MiB 이하의 메모리를 사용한다고 가정해도 좋다.

## 부분문제

번호	배점	제한
1	7	$N = 5$
2	8	$N \leq 100$
3	10	$0 \leq i < N$ 이며 $A[i] > 0$ 을 만족하는 $i$ 는 최대 하나이다.
4	30	$0 \leq i < \frac{N}{2}$ 인 모든 정수 $i$ 에 대해 $A[i] = 0$
5	45	추가적인 제약 조건이 없다.

## 채점 방식

### 부분문제 1, 2

1, 2번 부분문제에서는 `sorting` 함수의 반환값이 올바른 경우 부분문제 배점의 100%를 얻는다.

### 부분문제 3, 4, 5

3, 4, 5번 부분문제에서 응시자가 획득하는 점수는 다음과 같이 정해진다.

만약 프로그램이 비정상적으로 종료했거나, `sorting` 함수의 반환값이 올바르지 않은 경우 0점을 받는다.

그렇지 않은 경우, 해당 부분 문제의 점수는 다음과 같이 계산된다.

$Q_{max}$ 를 해당 부분문제에서 `sorting` 함수가 한 번 실행될 때 `ask_question` 함수가 호출된 횟수의 최댓값이라 하자.

$X$ 를  $Q_{max}$ 에 따라 다음과 같이 정의한다.

조건	$X$
$10\,000 < Q_{max}$	0
$80 < Q_{max} \leq 10\,000$	$90 - 35 \log_{10} \left( \frac{Q_{max}}{80} \right)$
$70 < Q_{max} \leq 80$	$170 - Q_{max}$
$Q_{max} \leq 70$	100

응시자는 해당 부분문제 점수의  $X\%$ 를 받는다.

## 예제

$N = 6$ 이고 엘리스가 가진 물건의 가치를 나타내는 배열  $A$ 가  $[5, 3, 3, 0, 8, 1]$ 인 경우를 생각해 보자.

그레이더는 처음에 아래 함수를 호출한다.

```
sorting(6)
```

참가자의 코드가 다음과 같이 상호작용할 수 있다.

```
ask_question([[0, 1], [1, 2], [2, 3], [3, 4], [4, 5]])
ask_question([[0, 1], [0, 2], [0, 3], [0, 4], [0, 5]])
```

첫 번째 호출의 경우, 엘리스가 0, 2, 4번 물건을 고를 경우 가치의 합이  $5 + 3 + 8 = 16$ 으로 최대가 된다. 따라서 이 호출은  $[1, 0, 1, 0, 1, 0]$ 을 반환한다.

두 번째 호출의 경우, 엘리스가 조건을 만족하면서 고를 수 있는 물건의 집합은  $\{1, 2, 4, 5\}$ 와  $\{1, 2, 3, 4, 5\}$ 가 있다. 따라서 이 호출은  $[0, 1, 1, 0, 1, 1]$ 이나  $[0, 1, 1, 1, 1, 1]$  중 하나를 반환한다.

다음과 같은 상호작용을 생각해 보자.

```
ask_question([[0, 1], [2, 3], [4, 5]])
ask_question([[0, 1], [1, 2], [2, 3], [3, 0], [4, 5]])
```

첫 번째 호출의 경우, `threads` 배열의 크기가  $N - 1$ 이 아니므로 유효한 호출이 아니다.

두 번째 호출의 경우, 0번 물건에서 4번 물건으로 실을 따라 이동할 수 없으므로 유효한 호출이 아니다.

조건을 만족하는 정수 배열  $P$ 는 다음 두 가지가 있다.

- $[3, 5, 1, 2, 0, 4]$
- $[3, 5, 2, 1, 0, 4]$

따라서 함수는 두 배열 중 하나를 반환해야 한다.

## 샘플 그레이더

샘플 그레이더의 입력 형식은 다음과 같다.

- line 1:  $N$
- line 2:  $A[0] A[1] \dots A[N - 1]$

제공되는 샘플 그레이더는 입력되는  $A[i]$ 가 0 이상  $10^9$  이하의 정수인 경우에만 정상 동작함이 보장된다.

샘플 그레이더는 여러분의 코드가 `sorting` 함수에서 반환한 배열과 `ask_question` 함수를 호출한 횟수를 다음 형식으로 출력한다.

- line 1: `sorting` 함수가 길이가  $M$ 인 배열  $P$ 를 반환했다고 할 때,  $P[0] P[1] \dots P[M - 1]$
- line 2: `ask_question` 함수를 호출한 횟수  $Q$

샘플 그레이더는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.