

# Report

## 프로그래머스 H-Index 문제(42747)



과목 : 코딩테스트지도

담당교수 : 이승진 교수님

학부 : IT융합자율학부

학번 : 202114136

이름 : 장준희

제출일 : 2025-03-23

작업 git url: <https://github.com/jjune960/coding-assignments/>

## 문제 설명

H-Index는 과학자의 생산성과 영향력을 나타내는 지표입니다. 어느 과학자의 H-Index를 나타내는 값인  $h$ 를 구하려고 합니다. 위키백과<sup>1</sup>에 따르면, H-Index는 다음과 같이 구합니다.

어떤 과학자가 발표한 논문  $n$ 편 중,  $h$ 번 이상 인용된 논문이  $h$ 편 이상이고 나머지 논문이  $h$ 번 이하 인용되었다면  $h$ 의 최댓값이 이 과학자의 H-Index입니다.

어떤 과학자가 발표한 논문의 인용 횟수를 담은 배열 `citations`가 매개변수로 주어질 때, 이 과학자의 H-Index를 return 하도록 `solution` 함수를 작성해주세요.

## 제한사항

- 과학자가 발표한 논문의 수는 1편 이상 1,000편 이하입니다.
- 논문별 인용 횟수는 0회 이상 10,000회 이하입니다.

## 요구 사항

논문  $n$ 편 중,  $h$ 번 이상 인용된 논문이  $h$ 편 (최댓값을 구하는 것이므로 사실상)

### 1차 시도 (강의자료 내림차순 정렬 참고)

#### 1. citations 내림차순 정렬 준비

H-Index를 구하려면 논문의 인용 횟수를 내림차순으로 정렬해야 한다.

원소의 값과 인덱스를 비교하여 최댓값을 찾아야 하기 때문이다.

하지만 `int` 배열은 기본적으로 내림차순 정렬을 지원하지 않는다.

따라서 `Integer` 객체 배열로 변환한 후 정렬해야 한다.

#### 2. Integer 배열로 변환하는 방법

`IntStream`을 활용하여 `int` 배열을 `Integer` 객체 배열로 변환할 수 있다.

코드: `Arrays.stream(citations).boxed();`

위 코드에서 `.boxed()`를 사용하면 `int` 타입의 배열이 `Integer` 객체 배열로 변환된다.

### 3. citations 내림차순 정렬 및 int 배열로 환원

내림차순 정렬은 Integer 스트림에서 sorted 명령어에 다음과 같은 옵션을 넣는다.

```
.sorted(Collections.reverseOrder())
```

이제 int 배열로 변환한다.

```
.mapToInt(i -> i).toArray();
```

이 값을 citations에 다시 정렬하면 준비가 완료된다.

### 4. h 값 찾기

For 문으로 원소의 값과 index가 같은지 비교한다.

코드:

```
for (int i = 0; i < citations.length; i++){  
    if(citations[i] == i)  
    {  
        answer = i;  
        break;  
    }  
}
```

이렇게 하면 h값이 나오는 코드를 구현하였다.

### 실행결과 (실패)

예상과 달리 citations = [3, 0, 6, 1, 5] 상태에서 h값이 3이 아닌 0이 반환되었다.

## 2차 시도 (GPT 참고 일부 수정)

실패 원인

### 1. 비교값 적용 착오 (h)

논문의 인덱스가 0부터 시작하기 때문에, 실제 논문의 개수(편수)를 나타내려면  $i+1$ 로 계산해야 합니다.

즉, 내림차순 정렬된 상태에서  $i$ 번째 논문에 대해 " $i+1$ 번 이상 인용된 논문이  $i+1$ 편 이상"이 되어야 하므로, 비교값은  $i+1$ 이 됩니다

수정한 코드

```
if(citations[i] == i + 1)
```

### 2. == 사용 안됨

처음에는  $\geq$ 와  $\leq$ 의 합집합이  $==$ 라고 생각할 수 있지만, H-Index의 조건은 " $i+1$ 번 이상 인용된 논문이  $i+1$ 편 이상"임을 만족하는 최대  $h$ 를 구하는 문제입니다.

만약  $citations[i] == i + 1$ 으로 조건을 체크하면, 정확히  $i+1$ 번 인용된 경우만 고려하게 되어, 인용 횟수가  $i+1$ 보다 많은 논문들도 조건에 포함시키지 못합니다.

예를 들어,  $citations = [3, 0, 6, 1, 5]$  상태에서, 정렬 후  $[6, 5, 3, 1, 0]$ 가 된다고 할 때,

"4번 이상 인용된 논문이 4편 이상"인 경우가 존재할 수 있지만,  $==$  조건은 인용 횟수가 정확히 4번인 경우만 체크하므로 적절한 H-Index 값을 찾지 못하게 됩니다.

따라서 H-Index 계산에서는  $citations[i] \geq i + 1$  조건을 사용하여, 해당 논문이 최소  $i+1$ 번 이상 인용되었는지 확인해야 합니다.

수정한 코드

```
if(citations[i] >= i + 1)
```

### 3. break 문 else로 이동

인덱스와 정확히 일치하는 경우만 찾으려고 했던 것만 생각해서 break문을

$citations[i] \geq i + 1$ 을 만족하면 바로 실행하게 한 것이다.

하지만 그렇게 한다면 조건이  $h+1$ 번째 인덱스에 정확히 존재해야만 return이 되는 문제

가 발생한다.

예를 들어 [6,5,4,1,0] 같은 경우에 값이 없는 문제가 발생한다. (4가 return되어야 함)

그래서 break 문은 citations[i] >= i + 1을 만족하지 않을 때 실행해야 하는 것이다.

수정한 코드

```
if (citations[i] >= i + 1) { // h번 “이상” 인용 된 논문을 검사
    answer = i + 1;
} else {
    break; // 조건을 만족하지 않으면 종료
}
```

### 실행결과 (성공)

Citations = [3, 0, 6, 1, 5] 일때 3이 return되는 올바른 값을 도출해 내었다.

### 배운 점

처음에는 h값이 값과 맞물리는 인덱스를 찾는 것에만 집중했다.

하지만 이러한 방식으로 풀기를 시도하면 오직 맞물리는 것만 추적하기 때문에 값이 없는 경우가 발생할 수 있다.

문제를 너무 편협하게 생각한 것이 문제였다.

앞으로는 다양한 경우의 수를 감안하여 문제를 풀어낼 것이다.