**[Algorithm] 27강 : 이진 탐색 기초 문제 풀이 — 나무늘보의 개발 블로그**

**노트북:** 첫 번째 노트북
**만든 날짜:** 2020-11-16 오후 10:10
**URL:** https://continuous-development.tistory.com/190?category=736684

Algorithm

# [Algorithm] 27강 : 이진 탐색 기초 문제 풀이

2020. 11. 13. 11:01 수정 삭제 공개

# <문제> 떡볶이 떡 만들기

## 〈문제〉 떡볶이 떡 만들기: 문제 설명

- 오늘 동빈이는 여행 가신 부모님을 대신해서 떡집 일을 하기로 했습니다. 오늘은 떡볶이 떡을 만드는 날입니다. 동빈이네 떡볶이 떡은 재밌게도 떡볶이 떡의 길이가 일정하지 않습니다. 대신에 한 봉지 안에 들어가는 떡의 총 길이는 절단기로 잘라서 맞춰줍니다.
- 절단기에 **높이(H)**를 지정하면 줄지어진 떡을 한 번에 절단합니다. 높이가 H보다 긴 떡은 H 위의 부분이 잘릴 것이고, 낮은 떡은 잘리지 않습니다.
- 예를 들어 높이가 19, 14, 10, 17cm인 떡이 나란히 있고 절단기 높이를 15cm로 지정하면 자른 뒤 떡의 높이는 15, 14, 10, 15cm가 될 것입니다. 잘린 떡의 길이는 차례대로 4, 0, 0, 2cm입니다. 손님은 6cm 만큼의 길이를 가져갑니다.
- 손님이 왔을 때 요청한 총 길이가 M일 때 **적어도 M만큼의 떡을 얻기 위해 절단기에 설정할 수 있는 높이의 최댓값을 구하는 프로그램**을 작성하세요.

난이도 ●●○ | 풀이 시간 40분 | 시간제한 2초 | 메모리 제한 128MB

**입력 조건**
- 첫째 줄에 떡의 개수 N과 요청한 떡의 길이 M이 주어집니다. (1 ≤ N ≤ 1,000,000, 1 ≤ M ≤ 2,000,000,000)
- 둘째 줄에는 떡의 개별 높이가 주어집니다. 떡 높이의 총합은 항상 M 이상이므로, 손님은 필요한 양만큼 떡을 사갈 수 있습니다. 높이는 10억보다 작거나 같은 양의 정수 또는 0입니다.

**출력 조건**
- 적어도 M만큼의 떡을 집에 가져가기 위해 절단기에 설정할 수 있는 높이의 최댓값을 출력합니다.

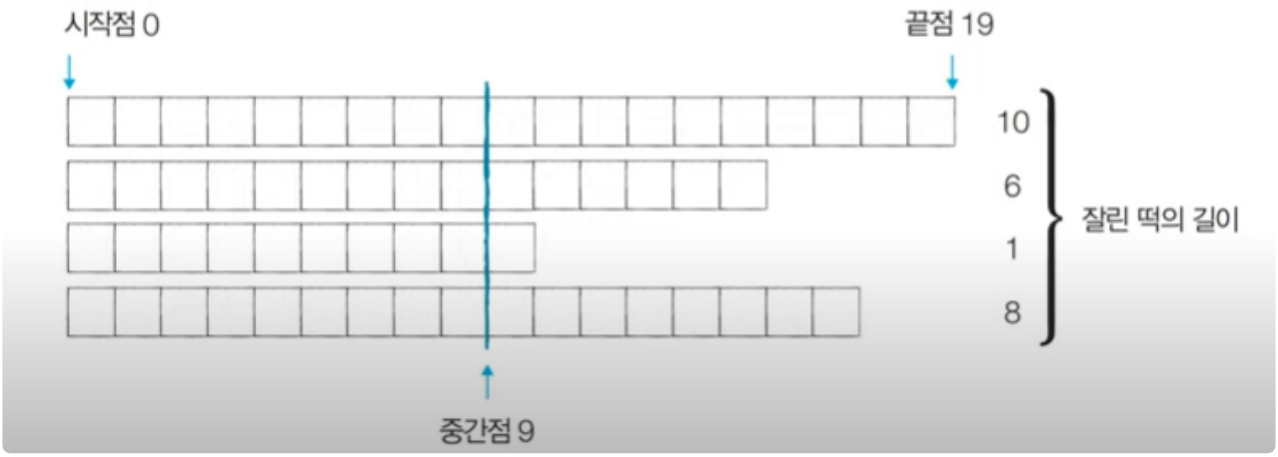
**입력 예시**
```
4 6
19 15 10 17
```

**출력 예시**
```
15
```

## #문제 해결 아이디어

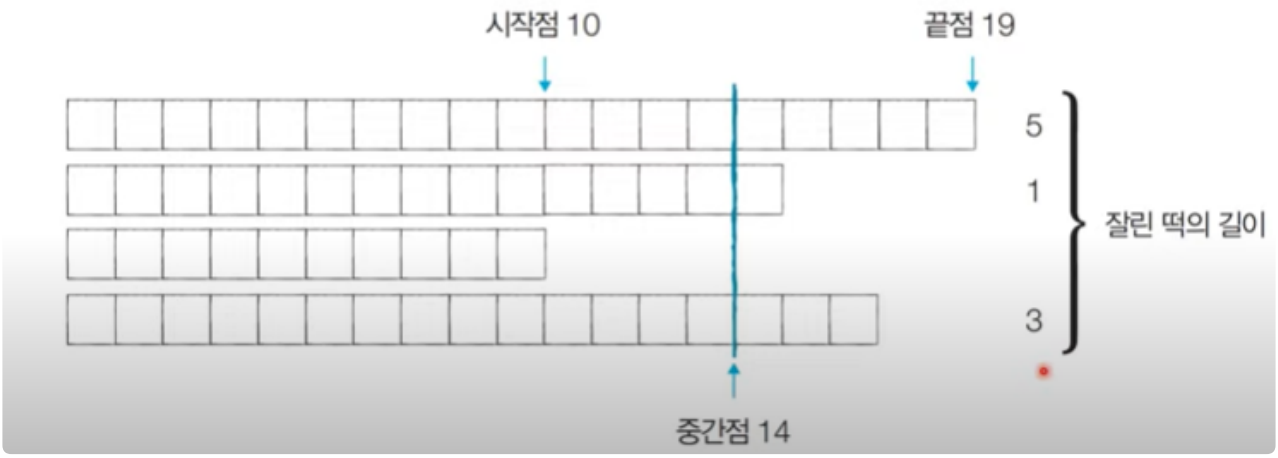
적절한 높이를 찾을 때까지 이진 탐색 수행하여 높이 H를 반복 조정

- **[Step 1]** 시작점: 0, 끝점: 19, 중간점: 9 이때 필요한 떡의 크기: M = 6이므로, **결과 저장**




가장 긴 떡의 끝점을 end 로 잡고 중간 지점을 찾아서 자른다. 이때 떡의 크기를 보고 더 크면 오른쪽으로 작으면 왼쪽으로 이동한다.

- **[Step 2]** 시작점: 10, 끝점: 19, 중간점: 14 이때 필요한 떡의 크기: M = 6이므로, **결과 저장**




이 과정을 반복한다..

# 답안

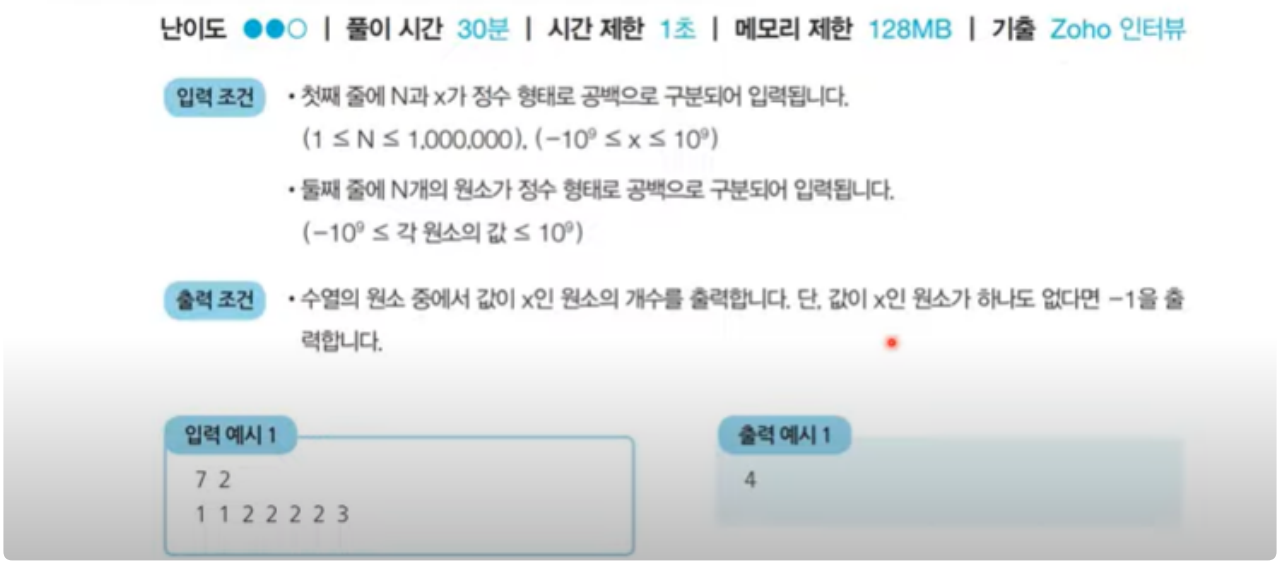
```
# 떡의 개수 N 과 요청한 떡의 길이 M을 입력
n,m = list(map(int,input().split(' ')))
# 각 떡의 개별 높이 정보를 입력
array = list(map(int,input().split()))

start = 0
end = max(array)

result = 0
end = max(array)

restult = 0

while(start <= end):
    totla = 0
    mid = (start+end) // 2
    for x in array:
    # 잘랐을때 떡의 양의 계산
        if x > mid:
            totla += x - mid
    # 떡의 양이 부족한 경우 덜 자르기
    if total < m:
        end = mid -1
    # 떡의 양이 충분한 경우 더 자르기
    else:
        result = mid # 최대한 덜 잘랐을 때가 정답이므로, 여기에 result에 기록
        start = mid + 1

print(result)
```

# <문제> 정렬된 배열에서 특정 수의 개수 구하기

## 〈문제〉 정렬된 배열에서 특정 수의 개수 구하기: 문제 설명

- N개의 원소를 포함하고 있는 수열이 오름차순으로 정렬되어 있습니다. 이때 이 수열에서 x가 등장하는 횟수를 계산하세요. 예를 들어 수열 {1, 1, 2, 2, 2, 2, 3}이 있을 때 x = 2라면, 현재 수열에서 값이 2인 원소가 4개이므로 4를 출력합니다.
- 단, 이 문제는 시간 복잡도 $O(logN)$으로 알고리즘을 설계하지 않으면 **시간 초과** 판정을 받습니다.

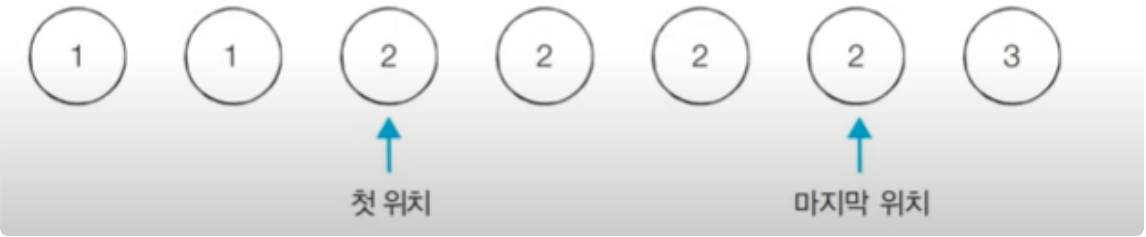
## 〈문제〉 정렬된 배열에서 특정 수의 개수 구하기: 문제 조건




## # 문제 해결 아이디어

시간 복잡도 O(log N)으로 동작하는 알고리즘을 요구하고 있다.
-일반적인 선형탐색으로는 시간 초과 판정을 받는다.
-하지만 데이터가 정렬되어 있어서 이진탐색이 가능하다,

특정값이 등장하는 첫 번째 위치와 마지막 위치를 찾아 위치 차이를 계산해 문제를 해결 할 수 있다.




## # 답안

```
from bisect import bisect_left , bisect_right


# 값을 찾아 데이터의 개수를 반환하는 함수
def count_by_range(array, left_value, right_value):
    right_index = bisect_right(array, right_value)
    left_index = bisect_left(array, left_value)
    return right_index - left_index

n , x = map(int,input().split()) # 데이터의 개수 N, 찾고자하는 값 x
array = list(map(int,inpit().split())) # 전체 데이터 입력받기

count = count_by_range(array, x ,x)

if count == 0:
    print(-1)
else:
    print(count)
```

---

www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvlEmdAghTr5mXQxGpHjWqSz0dgC
이 자료는 동빈 나 님의 이코 테 유튜브 영상을 보고 정리한 자료입니다.

'**Algorithm**' 카테고리의 다른 글□

[Algorithm] 28강 : 다이나믹 프로그래밍의 정의와 구현□

**[Algorithm] 27강 : 이진 탐색 기초 문제 풀이**□

[Algorithm] 26강 : 이진 탐색 알고리즘 정의와 구현□

[Algorithm] 25강 : 정렬 알고리즘 복잡도 비교 및 기본 문제□

[Algorithm] 24강 : 계수 정렬의 정의와 구현코드□

[Algorithm] 23강 : 퀵(quick) 정렬의 정의와 구현코드□

이진탐색 문제 이진탐색 에제

나아무늘보

혼자 끄적끄적하는 블로그 입니다.