

# 입국 심사

백준 3079

202021018 오윤주

# 문제

1 3079번

제출

맞은 사람

쏫코딩

재채점 결과

채점 현황

내 제출

난이도 기여

강의

질문 검색

입국심사

성공

다국어



한국어

1 실버 I

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	11794	2872	1947	26.240%

## 문제

상근이와 친구들은 오스트레일리아로 여행을 떠났다. 상근이와 친구들은 총  $M$ 명이고, 지금 공항에서 한 줄로 서서 입국심사를 기다리고 있다. 입국심사대는 총  $N$ 개가 있다. 각 입국심사관이 심사를 하는데 걸리는 시간은 사람마다 모두 다르다.  $k$ 번 심사대에 앉아있는 심사관이 한 명을 심사를 하는데 드는 시간은  $T_k$ 이다.

가장 처음에 모든 심사대는 비어있고, 심사를 할 준비를 모두 끝냈다. 상근이와 친구들은 비행기 하나를 전세내고 놀러가기 때문에, 지금 심사를 기다리고 있는 사람은 모두 상근이와 친구들이다. 한 심사대에서는 한 번에 한 사람만 심사를 할 수 있다. 가장 앞에 서 있는 사람은 비어있는 심사대가 보이면 거기로 가서 심사를 받을 수 있다. 하지만 항상 이동을 해야 하는 것은 아니다. 더 빠른 심사대의 심사가 끝나길 기다린 다음에 그 곳으로 가서 심사를 받아도 된다.

상근이와 친구들은 모두 컴퓨터 공학과 학생이기 때문에, 어떻게 심사를 받으면 모든 사람이 심사를 받는데 걸리는 시간이 최소가 될지 궁금해졌다.

예를 들어, 두 심사대가 있고, 심사를 하는데 걸리는 시간이 각각 7초와 10초라고 하자. 줄에 서 있는 사람이 6명이라면, 가장 첫 두 사람은 즉시 심사를 받으러 가게 된다. 7초가 되었을 때, 첫 번째 심사대는 비어있게 되고, 세 번째 사람이 그곳으로 이동해서 심사를 받으면 된다. 10초가 되는 순간, 네 번째 사람이 이곳으로 이동해서 심사를 받으면 되고, 14초가 되었을 때는 다섯 번째 사람이 첫 번째 심사대로 이동해서 심사를 받으면 된다. 20초가 되었을 때, 두 번째 심사대가 비어있게 된다. 하지만, 여섯 번째 사람이 그 곳으로 이동하지 않고, 1초를 더 기다린 다음에 첫 번째 심사대로 이동해서 심사를 받으면, 모든 사람이 심사를 받는데 걸리는 시간이 28초가 된다. 만약, 마지막 사람이 1초를 더 기다리지 않고, 첫 번째 심사대로 이동하지 않았다면, 모든 사람이 심사를 받는데 걸리는 시간이 30초가 되게 된다.

상근이와 친구들이 심사를 받는데 걸리는 시간의 최솟값을 구하는 프로그램을 작성하시오.

# 문제

예제 입력 1 복사

```
2 6
7
10
```

예제 출력 1 복사

```
28
```

첫번째, 두번째 사람이 입국심사를 받은 뒤  
7초에 세번째 사람, 10초에 네번째 사람  
14초에 다섯번째 사람

20초에 여섯번째 사람이 입국심사를 받는다면 30초에 전체 심사 완료  
하지만 1초를 더 기다려서 21초에 받는다면 28초에 전체 심사 완료

# 구현

1. 입국 심사대의 시간 정렬
2. Mid값은 심사하는데 걸리는 최소 시간
3.  $\text{Mid값} // \text{시간} = \text{심사할 수 있는 사람 수가 입력한 사람 수보다 크다면 mid값을 결과값으로}$
4. 아니라면 범위를 조정해서 다시 이분탐색

# 구현

```
num = list(map(int, input().split())) // N, M 입력받기

N = num[0]
M = num[1]
T = []

for i in range(N): // T에 시간 입력받기
    T.append(int(input()))

T.sort() // 시간 정렬
```

# 구현

```
def bsort(T, num):  
    low = 1  
    high = T[0] * num // low, high 설정  
  
    while(low <= high):  
        count = 0  
        mid = (low + high) // 2 // mid값 설정  
  
        for i in T:  
            count += mid // i // count에 mid // 시간 한 값 더해주기  
            = 심사 가능한 사람 수  
  
        if(num <= count):  
            result = mid  
            high = mid - 1 // 가능하다면 mid값이 결과값, high값 조정해주기  
        else:  
            low = mid + 1 // 아니라면 low값 조정  
  
    return result  
  
print(bsort(T, M)) // 결과 출력
```

**감사합니다.**