



입국심사

🔖 태그	프로그래머스
📅 날짜	@2024년 12월 11일
☑ 상태	살짝 참고했따
🔗 참고링크	https://gnuoyus.tistory.com/93

아이디어



모든 사람이 심사를 받는데 걸리는 시간의 최소값을 `return` 하도록 `solution` 함수 작성

이분 탐색으로 입국 심사를 완료하는 데 필요한 예상 시간을 탐색하고,

그 시간 동안 `n` 명을 모두 심사할 수 있는지 확인

그 중 최소 시간을 반환

1. `min` (입국 심사 최소 시간): 1

`max` (입국 심사 최대 시간): `(long) times[times.length-1] * (long) n` 으로 설정

2. `binarySearch(min, max, n, times)` 함수

a. `min + max / 2` 를 예상 시간으로 두고

b. `times` 배열을 돌면서 해당 예상 시간 동안 각 심사관이 처리할 수 있는 사람 수 계산

c. 그 수가 `n` 보다 많으면 `answer` 갱신 + `max` 이동으로 범위 좁히기

d. `n` 보다 적으면 `min` 이동으로 범위 좁히기

주의)

1. `n`, `times` 배열의 원소 값이 매우 큼!

∴ `int` 가 아닌 `long` 배열 사용 필요

2. 종료조건을 `left <= right` 로 해야함!

`left < right` 로 하면 6번 테스트케이스 틀림

문제를 일으킬 수 있는 예외적인 테스트 케이스

• $n = 1, arr = [1]$

1. `left < right` 의 경우:

• $left = 1, right = 1$

• 루프가 실행되지 않고 종료 → 결과가 계산되지 않을 가능성.

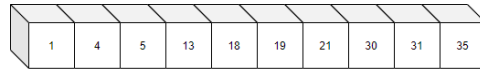
2. `left <= right` 의 경우:

• 루프가 한 번 실행되며 `mid = 1` 을 계산 → 올바른 결과 반환.

관련 알고리즘

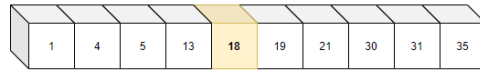
이분 탐색(Binary Search)

0.오름차순으로 정렬된 배열



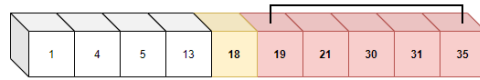
31이란 데이터를 탐색한다고 하자.
1. 배열의 중앙값을 선택한다. (배열의 크기 10)

low : 0
high : 9
mid : 4



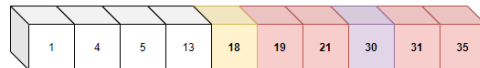
2. 18(중앙값)과 31을 비교한다.
31보다 18이 작으므로 우측요소를 선택한다.

low : 5
high : 9



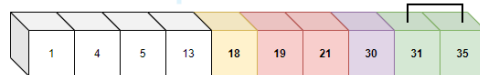
3. 선택된 우측 요소의 중앙값을 선택한다.

low : 5
high : 9
mid : 7



4. 30(중앙값)과 31을 비교한다.
31보다 30이 작으므로 우측요소를 선택한다.

low : 8
high : 9



5. 남은 배열중에 좌측요소(31)를 선택하여 31과 비교한다.
31과 동일하면 탐색 종료

low : 8
high : 9
mid : 8

