

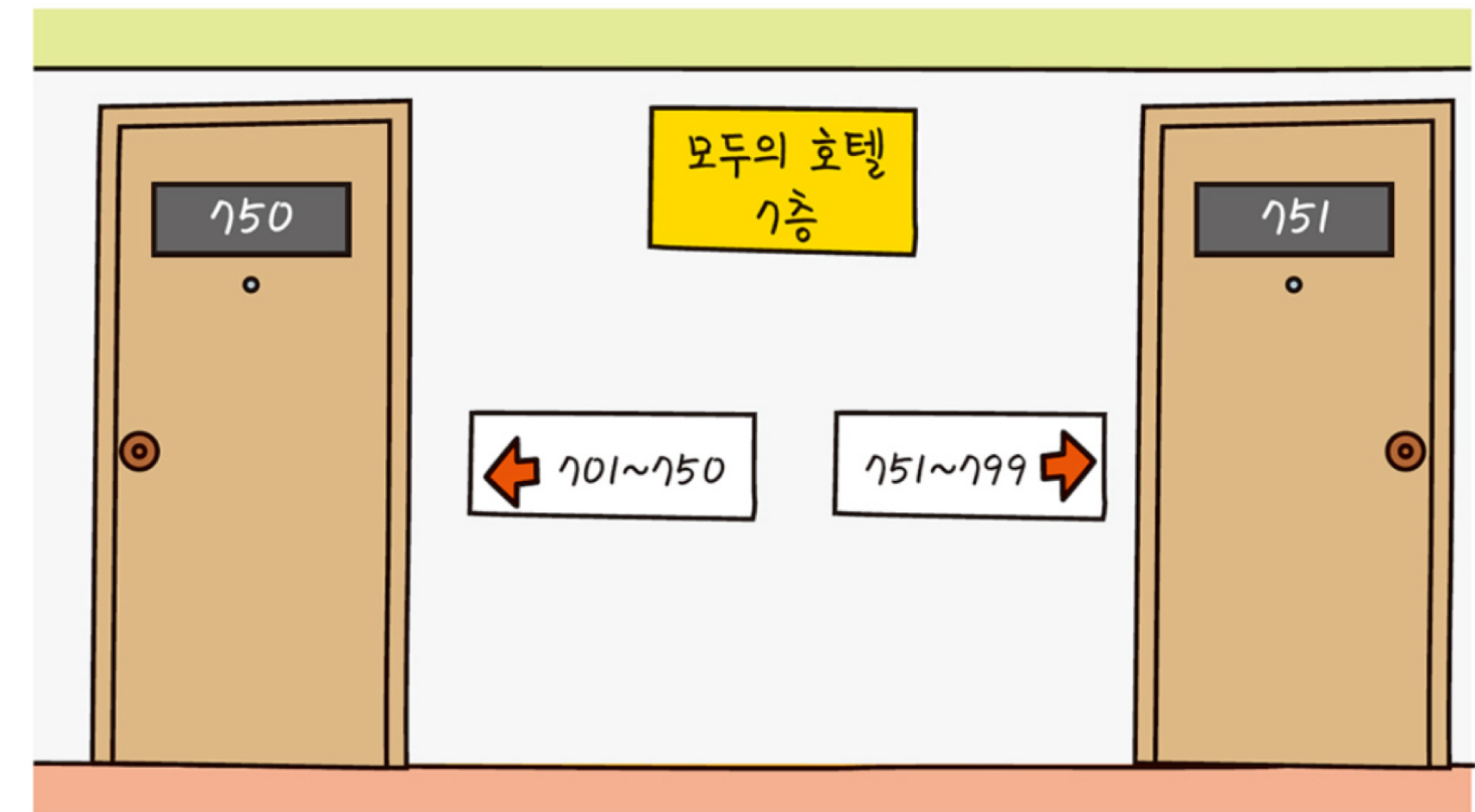
이분 탐색

모두의 알고리즘 with 파이썬 - 문제 12

20.06.14 장예훈

이분 탐색

이분 탐색 (Binary Search) 의 '이분'은 '둘로 나눈다'



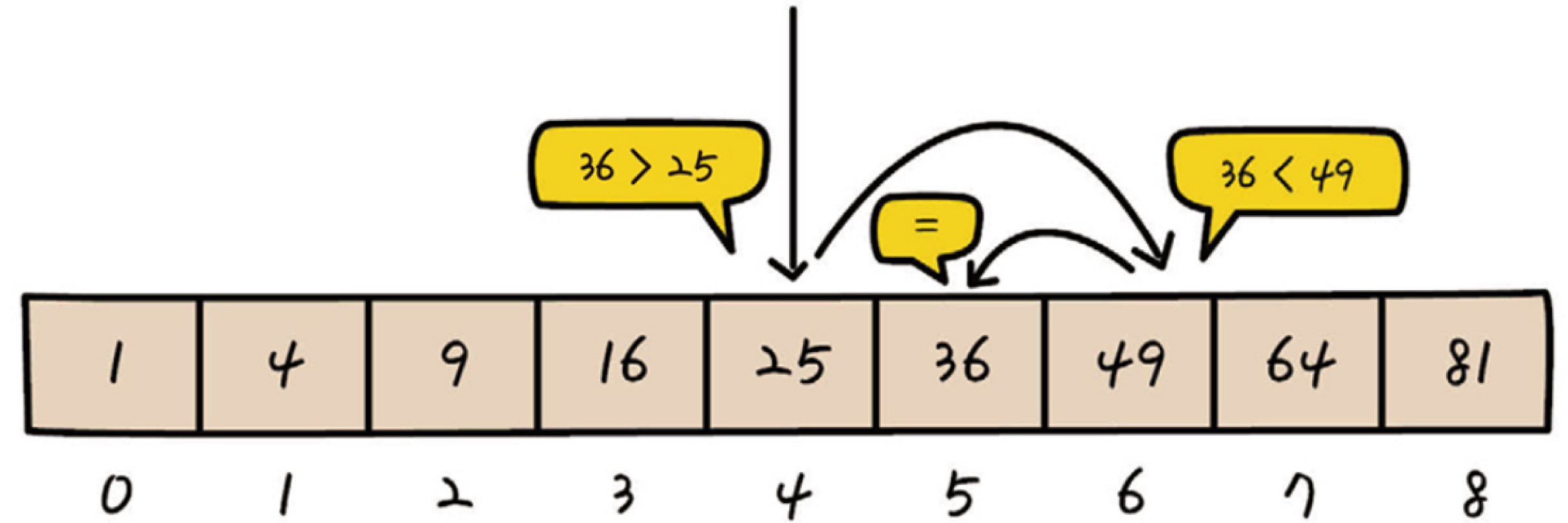
적당한 기준을 잡고 찾고자 하는 값과 비교 후 그 값이 있을 방향으로만 다시 탐색

!! 탐색하는 대상이 정렬되어 있어야 이분 탐색 가능 **!!**

이분 탐색 알고리즘

입력: [1, 4, 9, 16, 25, 36, 49, 64, 81]

찾는 값: 36



1. 리스트의 중간 위치 찾기
2. 찾는 값과 중간 위치 값 비교
3. 찾는 값 > 중간 위치 값 ➡ 리스트의 오른쪽 부분 탐색 / 찾는 값 < 중간 위치 값 ➡ 리스트의 왼쪽 부분 탐색
4. 원하는 값을 찾을 때까지 1~3 반복

이분 탐색 알고리즘

```
1 def binary_search(a, x):
2     start = 0
3     end = len(a) - 1
4     while start <= end:
5         mid = (start + end) // 2
6         if x == a[mid]:
7             return mid
8         elif x > a[mid]:
9             start = mid + 1
10        else:
11            end = mid - 1
12    return -1
```

```
d = [1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
print(binary_search(d, 36))
print(binary_search(d, 100))
```

```
5
-1
```

line 2-3: 탐색할 범위를 지정할 변수

line 4: 탐색할 범위가 남아있는 동안 반복

line 5: 탐색할 범위의 중간 위치

line 8-9: 찾는 값 > 중간 위치 값

➡ 오른쪽만 탐색하기 위해 start 값 수정

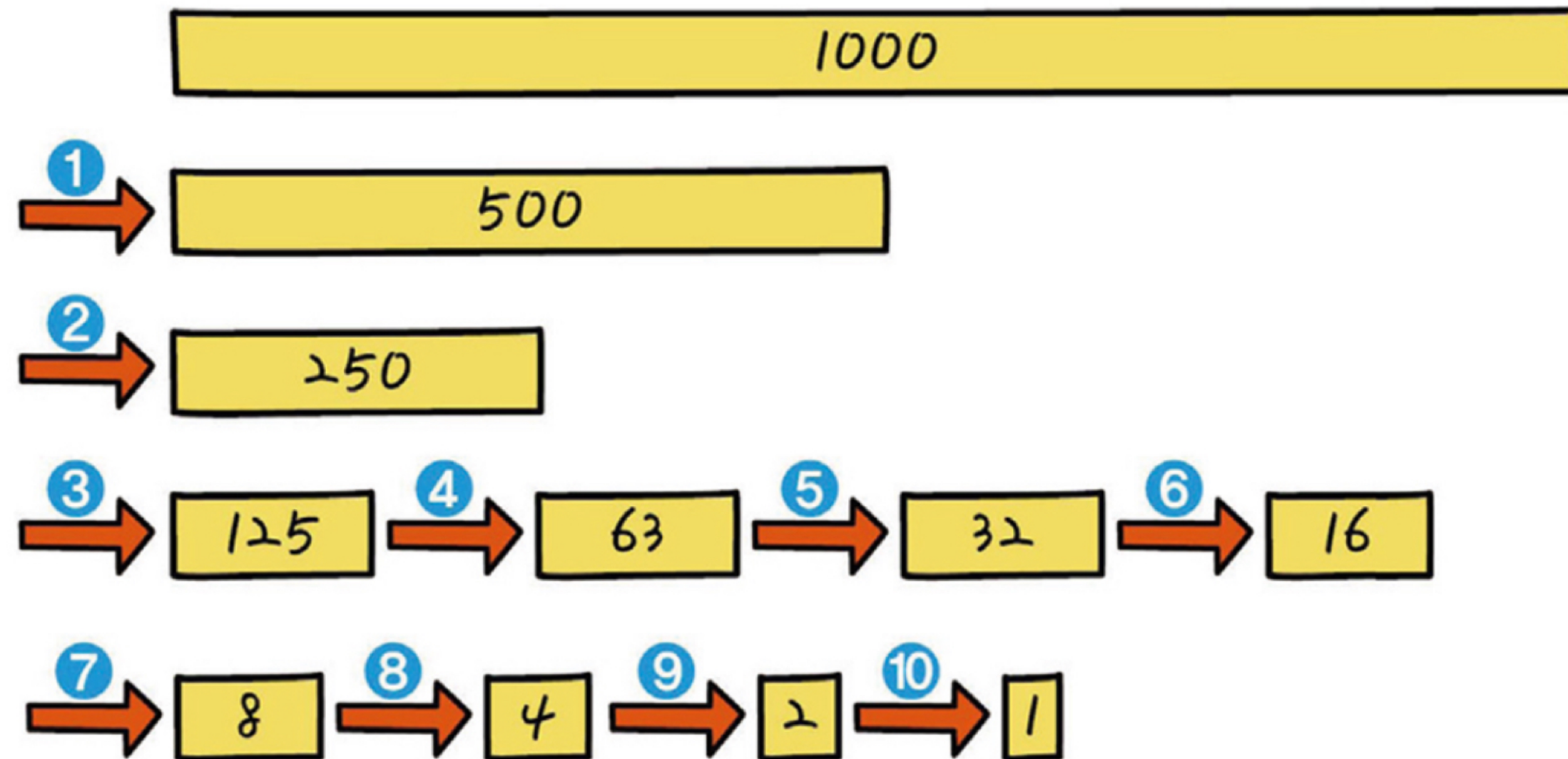
line 10-11: 찾는 값 < 중간 위치 값

➡ 왼쪽만 탐색하기 위해 end 값 수정

line 12: 찾는 값 없을 시 -1 출력

알고리즘 분석

계산 복잡도는 $O(\log_2 n)$



* 순차 탐색은 최악의 경우 1000번 비교

* 이분 탐색은 최악의 경우라도 10번 비교

($\because \log_2 1000 = 9.966$)

연습 문제

다음 과정을 참고하여 재귀 호출을 사용한 이분 탐색 알고리즘을 만들어 보세요.

- 1 주어진 탐색 대상이 비어 있다면 탐색할 필요가 없습니다(종료조건).
- 2 찾는 값과 주어진 탐색 대상의 중간 위치 값을 비교합니다.
- 3 찾는 값과 중간 위치 값이 같다면 결과값으로 중간 위치 값을 돌려줍니다.
- 4 찾는 값이 중간 위치 값보다 크다면 중간 위치의 오른쪽을 대상으로 이분 탐색 함수를 재귀 호출합니다.
- 5 찾는 값이 중간 위치 값보다 작다면 중간 위치의 왼쪽을 대상으로 이분 탐색 함수를 재귀 호출합니다.

연습 문제

다음 과정을 참고하여 재귀 호출을 사용한 이분 탐색 알고리즘을 만들어 보세요.

```
1 def binary_search_2(a, x, start, end):
2     if start > end:
3         return -1
4     mid = (start+end)//2
5     if x==a[mid]:
6         return mid
7     elif x>a[mid]:
8         return binary_search_2(a, x, mid+1, end)
9     else:
10        return binary_search_2(a, x, start, mid-1)
11
12    return -1
13
14 def bsearch(a, x):
15    return binary_search_2(a, x, 0, len(a)-1)
```

```
print(bsearch(d, 36))
print(bsearch(d, 100))
```

5
-1

- 1 주어진 탐색 대상이 비어 있다면 탐색할 필요가 없습니다(종료조건).
- 2 찾는 값과 주어진 탐색 대상의 중간 위치 값을 비교합니다.
- 3 찾는 값과 중간 위치 값이 같다면 결과값으로 중간 위치 값을 돌려줍니다.
- 4 찾는 값이 중간 위치 값보다 크다면
중간 위치의 오른쪽을 대상으로 이분 탐색 함수를 재귀 호출합니다.
- 5 찾는 값이 중간 위치 값보다 작다면
중간 위치의 왼쪽을 대상으로 이분 탐색 함수를 재귀 호출합니다.

감사합니다 😊