이진 탐색

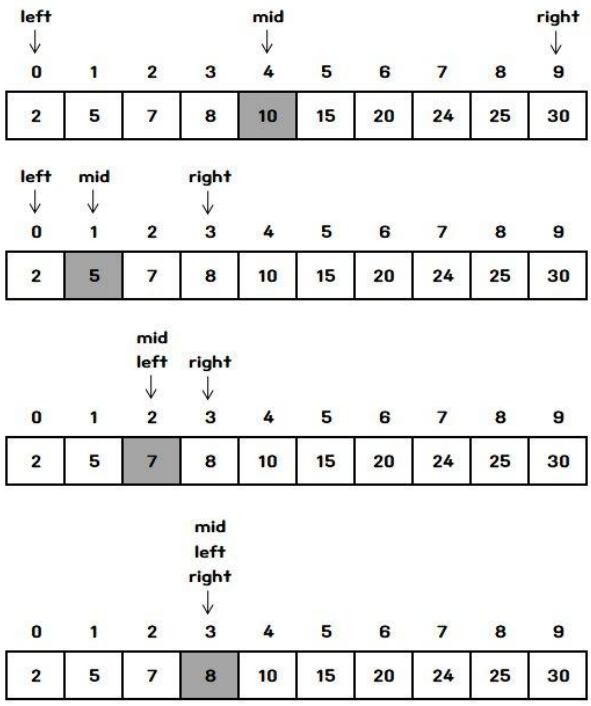
nums에 오름차순으로 정렬된 정수 배열이 주어지고, target에 nums배열에서 찾고자 하는 값이 주어지면 nums배열에서 target의 인덱스 번호를 찾아 반환하는 프로그램을 작성하세요.
인덱스 번호는 0번부터 시작합니다.
target값이 nums에 존재하지 않을 경우 -1를 반환합니다.

입출력 예:

nums	target	answer
[2, 5, 7, 8, 10, 15, 20, 24, 25, 30]	8	3
[-3, 0, 2, 5, 8, 9, 12, 15]	0	1
[-5, -2, -1, 3, 8, 9, 12, 17, 23]	2	-1
[3, 6, 9, 12, 17, 29, 33]	3	0
[1, 2, 3, 4, 5, 7, 9, 11, 12, 15, 16, 17, 18]	18	12

제한사항:
- nums의 길이는 100,000,000을 넘지 않습니다. nums의 원소는 유일값입니다.
- -100,000,000 <= nums[i] <= 100,000,000

예제 1번을 이진탐색으로 8을 찾는 과정:



lower bound search

찾고자 하는 값보다 크거나 같은 것 중에서 가장 작은 값을 찾아주는 이분검색방법

```
def solution(nums, weight):
    left = 0
    right = len(nums)
    while left < right:
        mid = (left + right) // 2
        if weight > nums[mid]:
            left = mid + 1
        else:
            right = mid
    return -1 if right == len(nums) else right

print(solution([100, 120, 150, 160, 165, 170, 175, 180, 190, 200], 180))
```

upper bound search

찾고자 하는 값보다 큰 것 중에서 가장 작은 값을 찾아주는 이분검색방법

```
def solution(nums, weight):
    left = 0
    right = len(nums)
    while left < right:
        mid = (left + right) // 2
        if weight >= nums[mid]:
            left = mid + 1
        else:
            right = mid
    return -1 if right == len(nums) else right

print(solution([100, 120, 150, 160, 165, 170, 175, 180, 190, 200], 175))
```

트럭 찾기

현수는 이사를 하려고 합니다.
이삿짐 센터에는 여러 개의 트럭이 있습니다. 각 트럭은 짐을 실을 수 있는 총 무게 제한이 있습니다.
이사 비용은 현수가 선택한 트럭의 무게 제한에 곱하기 10을 한 값입니다.
nums에 각 트럭의 무게 제한이 오름차순으로 주어지고, 현수의 이사하는 짐의 총 무게가 weight주어지면 현수가 짐을 실을 수 있는 최소비용의 트럭을 찾아 선택된 트럭의 인덱스 번호를 반환하는 프로그램을 작성하세요.
인덱스 번호는 0번부터 시작합니다.
현수가 이사할 수 있는 트럭이 존재하지 않을 경우 -1를 반환합니다.

입출력 예:

nums	weight	answer
[100, 120, 150, 160, 165, 170, 175, 180, 190, 200]	170	5
[200, 250, 260, 265, 275, 290, 300, 325, 350, 370]	270	4
[50, 55, 60, 65, 70, 80, 90]	80	5
[20, 30, 40, 50, 60, 70]	90	-1
[10, 30, 50, 70, 80, 90, 100, 110, 120]	115	8

제한사항:
• nums의 길이는 100,000,000을 넘지 않습니다. nums의 원소는 유일값입니다.
• 1 <= nums[i] <= 100,000,000

고정된 숫자

오름차순으로 정렬된 일차원 배열의 원소 중 인덱스 번호(인덱스 번호는 0번부터 시작합니다.)와 자기 자신의 값이 같으면 이 원소를 고정된 숫자라고 합니다.
예를 들어 [-3, -2, 0, 1, 3, 5, 8, 9, 12] 배열에서 고정된 숫자는 5입니다. 배열 원소 5의 인덱스 번호가 5로 원소와 인덱스 값이 같습니다.
매개변수 nums에 오름차순으로 정렬된 정수 배열이 주어지면 배열 원소 중 고정된 숫자를 찾아 반환하는 프로그램을 작성하세요.
고정된 숫자는 유일합니다. 고정된 숫자가 없다면 -1를 반환하세요.

입출력 예:

nums	answer
[-3, -2, 0, 1, 3, 5, 8, 9, 12]	5
[-10, -5, -2, 3, 4, 6, 7, 8]	3
[1, 2, 3, 4, 5, 6, 7, 8, 9]	-1
[-5, -3, 0, 1, 2, 3, 4, 7]	7
[0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]	0

제한사항:
• nums의 길이는 1,000,000을 넘지 않습니다. nums의 원소는 유일값입니다.
• 1 <= nums[i] <= 100,000,000