

## 알고리즘과게임콘텐츠

### 10장 자료탐색기법\_이진탐색

## 학습 목표

1. 이진탐색 알고리즘의 개념을 이해한다.
2. 분할 정복의 개념을 이해한다.
3. 분할정복 알고리즘의 종류를 구분할 수 있다.

# 이진탐색(BINARY SEARCH) 개념

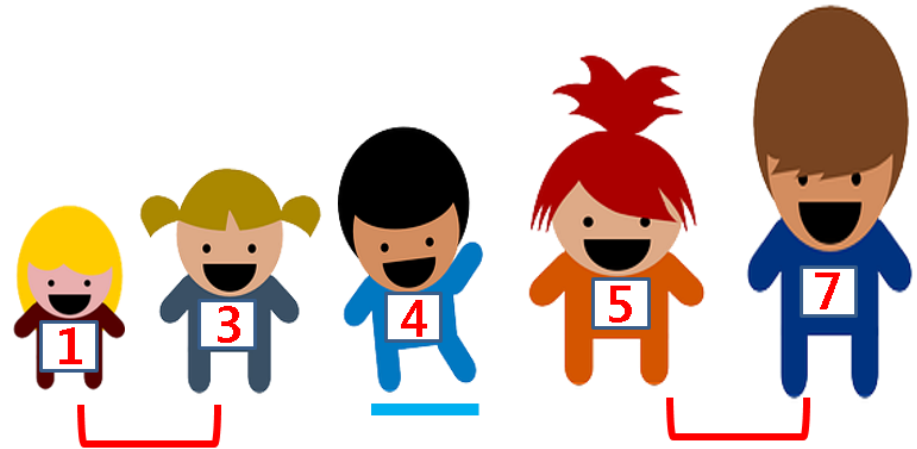
- 자료를 절반으로 나눈 후 찾는 값이 어느쪽에 위치해 있는지 파악해 탐색의 범위를 반으로 줄여가면서 원하는 값을 찾는 알고리즘

- 7번 선수 찾기

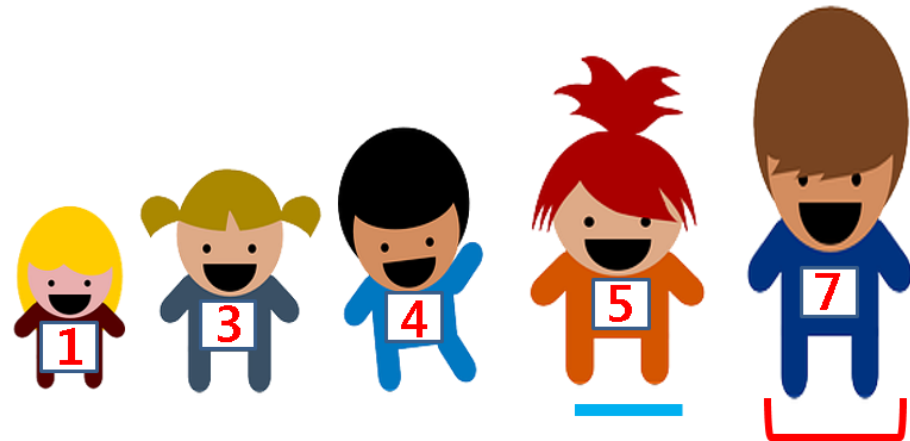
①



②



③



# 이진탐색(BINARY SEARCH) 개념

- 기본 전략

- 전체 데이터가 반드시 정렬되어져 있어야 하며,

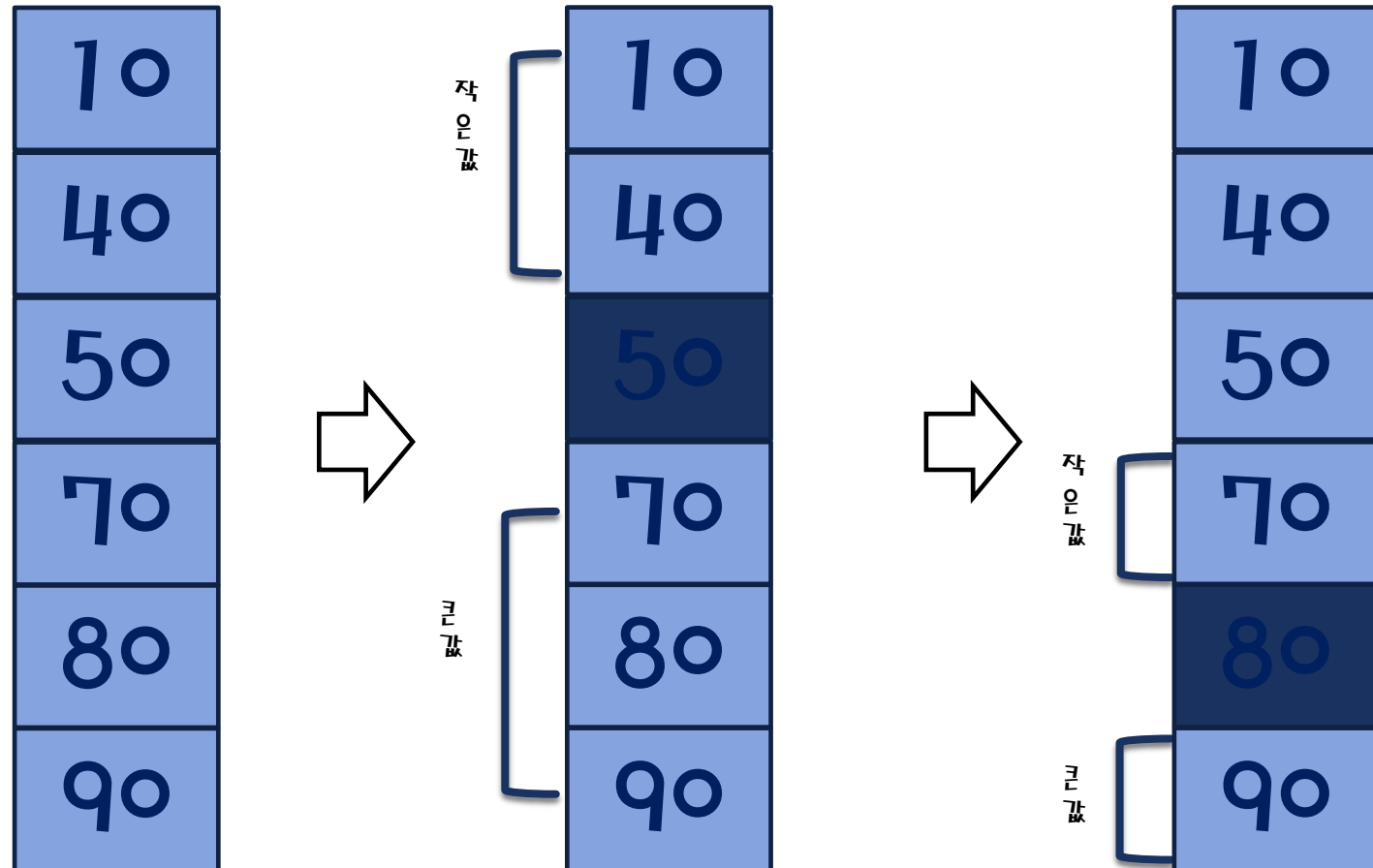
중간위치를 기준으로 왼쪽과 오른쪽으로 나누어 탐색함

- 오름차순으로 정렬된 경우 중간 값보다 작으면 왼쪽 탐색
- 오름차순으로 정렬된 경우 중간 값보다 크면 오른쪽 탐색
- 탐색의 범위를 반으로 줄여가며 원하는 값 탐색

# 이진탐색(BINARY SEARCH) 개념

- 기본 전략

- 90을 찾을 경우



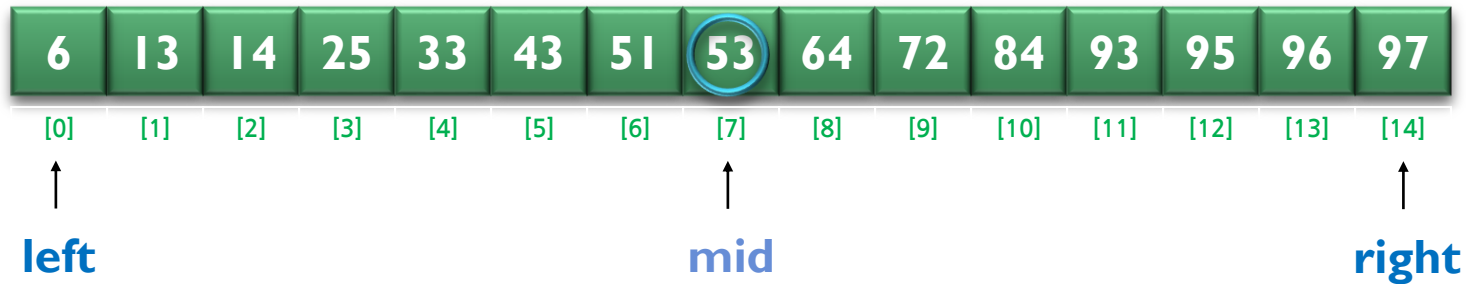
# 이진탐색(BINARY SEARCH) 개념

- 이진 탐색 (Binary Search)
  - 정렬된 배열  $a()$  에서 특정 값  $value$  의 인덱스  $i$  찾기
  - 반복
    - 조건을 만족 시키는 동안  $a(left) \leq value \leq a(right)$
- 예) 33을 찾기 위한 이진 탐색
  - 0번지의 값을  $left$ 로 마지막 번지의 값을  $right$ 로 지정

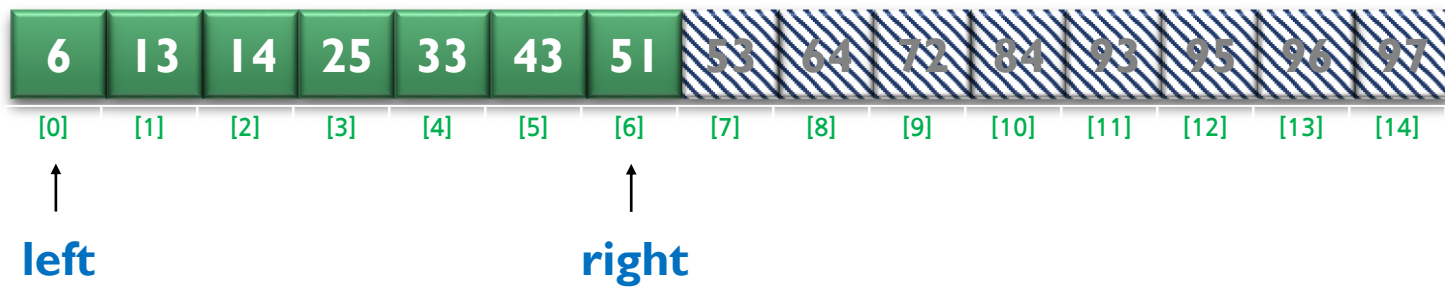


# 이진탐색(BINARY SEARCH) 개념

- 예] 33을 찾기 위한 이진 탐색
  - 가운데 번지( $14/2$ )를 mid로 지정

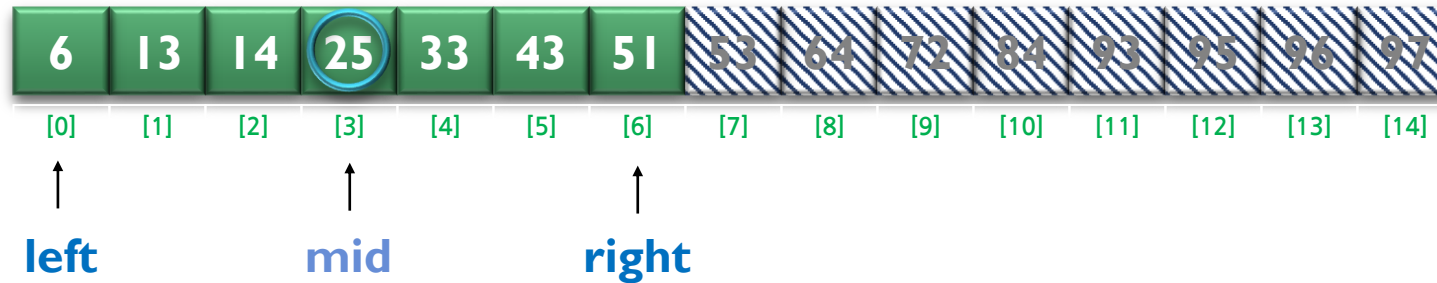


- mid의 값이 찾는 값보다 크면 mid-1번지의 값을 right로 지정

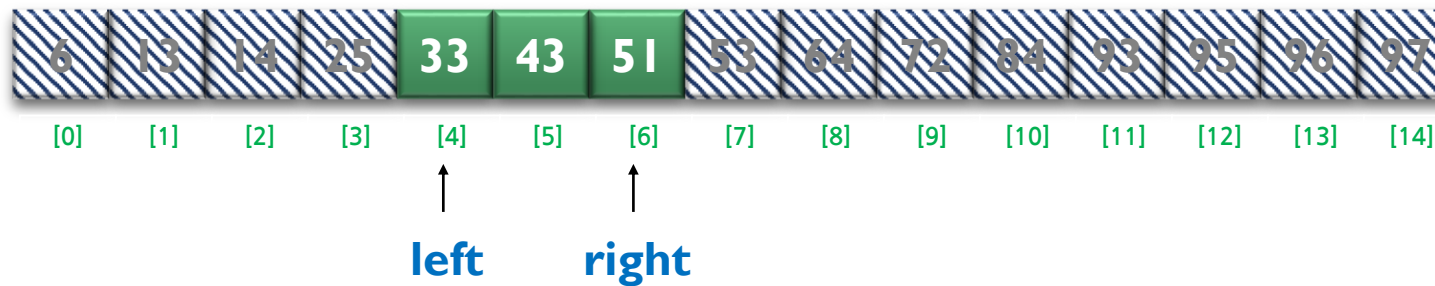


# 이진탐색(BINARY SEARCH) 개념

- 예] 33을 찾기 위한 이진 탐색
  - 반복하여 가운데 번지( $0+6/2$ )에 해당하는 3번지를 mid로 지정



- mid의 값이 찾는 값보다 작으면 mid+1번지의 값을 left로 지정



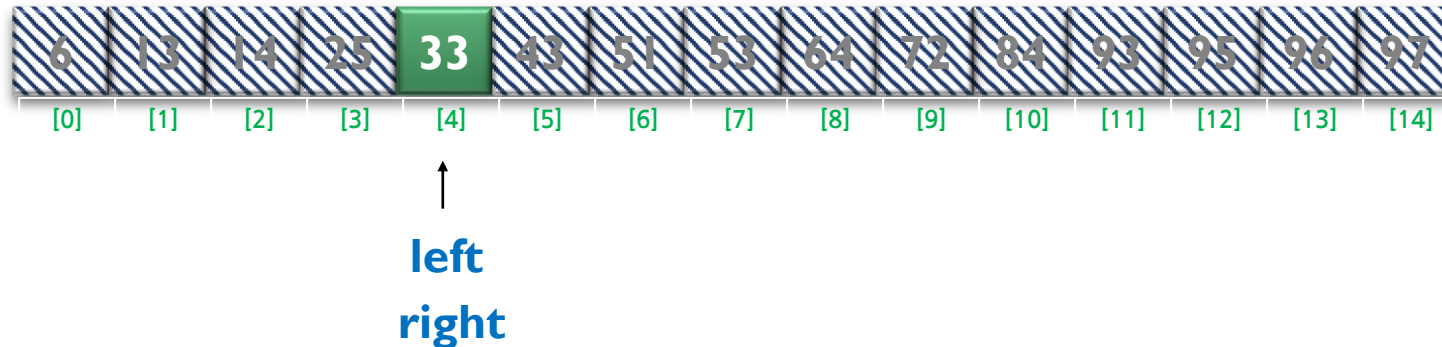


# 이진탐색(BINARY SEARCH) 개념

- 예] 33을 찾기 위한 이진 탐색
  - 반복하여 가운데 번지( $(4+6)/2$ )에 해당하는 5번지를 mid로 지정

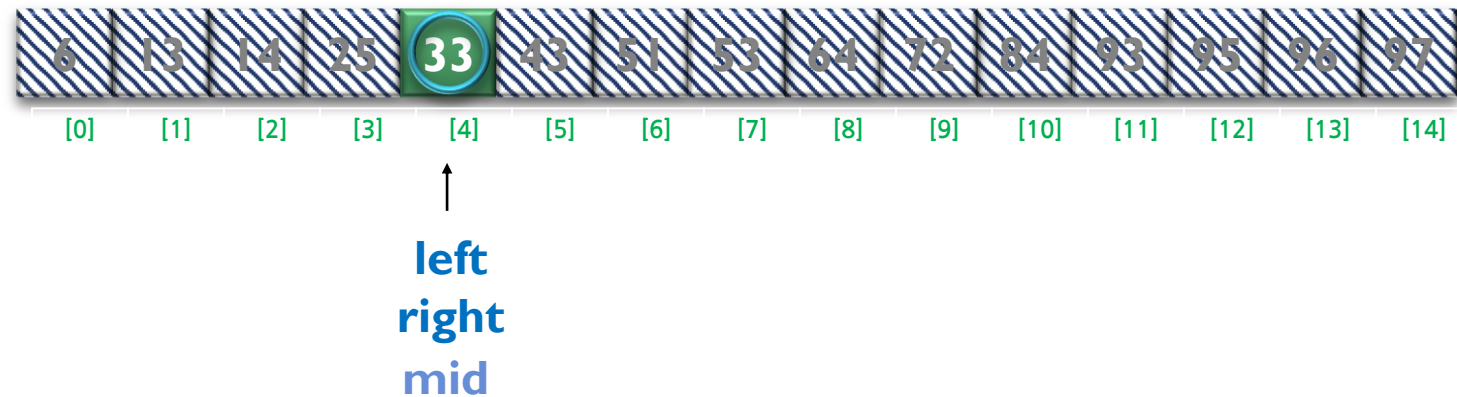


- mid의 값이 찾는 값보다 크면 mid-1번지의 값을 right로 지정

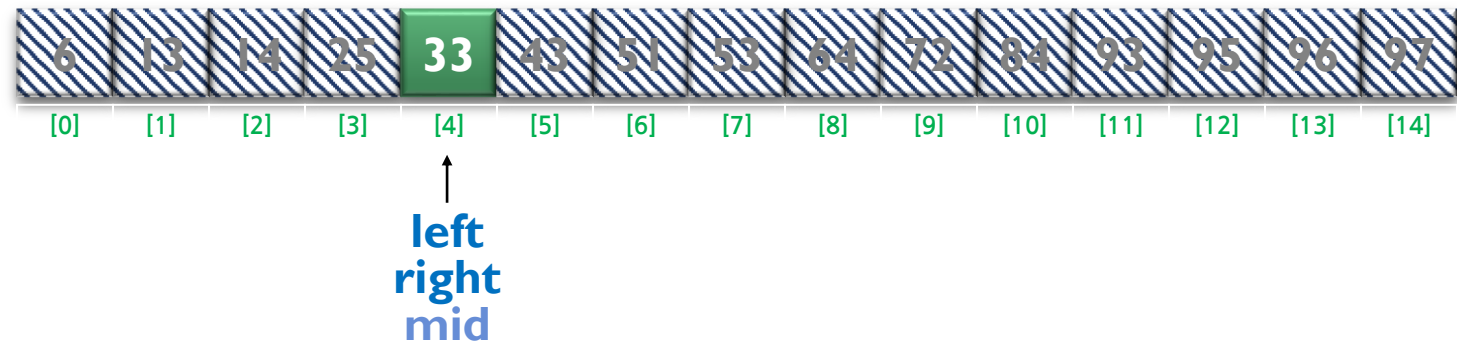


# 이진탐색(BINARY SEARCH) 개념

- 예] 33을 찾기 위한 이진 탐색
  - 반복하여 가운데 번지( $(4+4)/2$ )에 해당하는 4번지를 mid로 지정



- 모두가 모인 주소지가 찾고자 하는 값이므로 찾는 인덱스의 값은 4



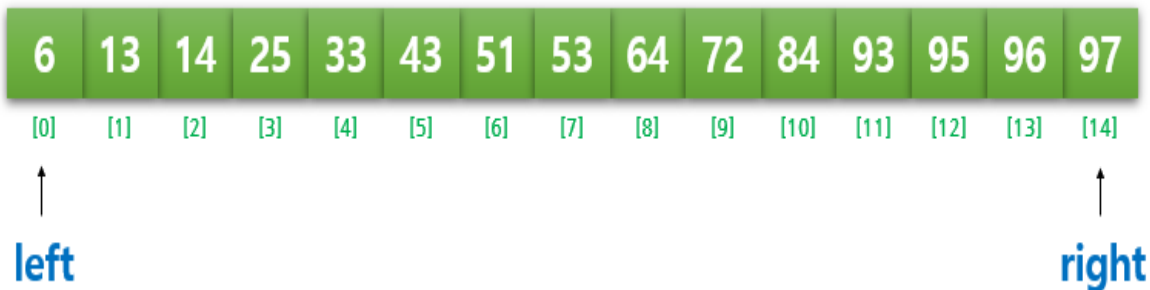
# 이진탐색(BINARY SEARCH) 개념

- 코딩

인덱스 0번째를 left에 마지막 번째를 right 로 설정합니다.

- left=0

- right=len(배열이름) -1



```
while (left <= right):  
    mid = (left + right) // 2  
    if target > data[mid]:  
        left = mid + 1  
    elif target < data[mid]:  
        right = mid - 1  
    else:  
        break  
  
if target == data[mid]:  
    print("찾는 수는 %d 번째 인덱스 위치에 있습니다"%(mid))  
else:  
    print("찾는 수가 존재하지 않습니다. ")
```

# 이진탐색(BINARY SEARCH) 개념

## ■ 이진탐색의 장단점

|    |  |
|----|--|
| 장점 | <ul style="list-style-type: none"><li>• 알고리즘이 간단하고 이해하기 쉽다.</li><li>• 자료의 개수가 많아도 사용할 수 있다.</li><li>• 비교 횟수가 적어 속도가 빠르다.</li></ul> |
| 단점 | <ul style="list-style-type: none"><li>• 자료가 미리 정렬 되어져 있어야 한다.</li></ul>  |

# 순차\_이진탐색

EBS1

순차탐색  
(선형탐색)

15 11 1 3 8

3

주어진 데이터를 처음부터  
차례대로 비교하면서 찾는 방법

동영상 3:55

# 자료 탐색 기법\_이진탐색

- 이진탐색의 개념
- 이진탐색의 알고리즘
- 분할정복의 개념
- 쿼정렬과 이진탐색

