알고리즘과게임콘텐츠

11장 효율적인 정렬과 탐색 알고리즘

- 분할정복 알고리즘의 이해와 방법 -

학습 목표

1. 분할 정복 알고리즘의 개념을 이해한다.

2. 실생활에서 분할 정복 알고리즘을 이용하여 문제해결을 할 수 있다.

3. 이진 탐색과 퀵 정렬을 분할 정복 방식으로 해결할 수 있다.

분할 정복 (DIVIDE-AND-CONQUER) 알고리즘 개념

- 주어진 문제의 입력을 분할하여 문제를 해결 (정복)하는 방식의 알고리즘
  - 분할된 입력에 대한 문제 : 부분문제 (sub-problem)
    - 부분문제는 더 이상 분할할 수 없을 때까지 계속 분할 과정을 반복함
  - 알고리즘을 적용한 부분문제의 결과 : 부분 결과 (sub-solution)
  - 분할한 입력에 대하여 동일한 알고리즘을 적용하여 부분 결과를 반복적으로 취함
  - 부분결과를 취합하여 원래 문제의 결과를 얻음
- 문제를 더 이상 나눌 수 없을 때까지 나누고, 이렇게 나누어진 문제들을 각각 풂으로써 결국 전체 문제의 답을 얻는 알고리즘

분할 정복 (DIVIDE-AND-CONQUER) 알고리즘 개념

- 문제를 쪼개는 요령이나 규칙은 없음. 개발자의 창의에 달려 있음.
  - 컴퓨팅 사고력 + 창의적 문제 해결 능력
- 알고리즘을 설계하는 요령

1. 분할(Divide) : 문제가 분할이 가능한 경우, 2개 이상의 하위 문제로 나눔

2. 정복(Conquer) : 하위 문제가 여전히 분할이 가능한 상태라면 하위 집합에 대해 1을 반복 수행,

   그렇지 않다면 하위 문제의 답 풀기

3. 결합(combine) : 2 과정에서 정복된 답을 결합

분할 정복 (DIVIDE-AND-CONQUER) 알고리즘의 예

- 문제 상황

400명의 유치원생을 위하여 포장한 동일한 선물 가운데 더러운 수건이 따라 들어간 선물 상자 찾아내야 함.

분할 정복 (DIVIDE-AND-CONQUER) 알고리즘의 예

- 해결 방법 1 : 단순하기 문제 풀기
  - 한 개씩 비교
  - 최악의 경우 399번의 비교

- 해결 방법 2 : 분할 정복 알고리즘
  - ½ 로 나누어 비교

분할 정복 (DIVIDE-AND-CONQUER) 알고리즘의 예

- 주차장에서의 방황

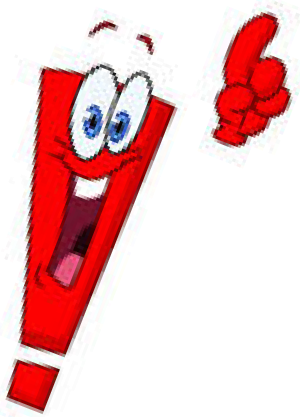
- 사전에서 단어 찾기

분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 정렬된 수들의 집합에서 특정 값을 확인하기
  - 정렬된 수

    1, 7, 14, 17, 26, 59, 63, 77, 79, 87, 88, 90, 92, 96, 98, 99

  - 96을 확인하고자 한다면?
    1. 앞에서부터 순서대로 하나씩 비교하며 확인
       - 무식하게 풀기(Brute Force)의 순차 검색
    2. 분할 정복
       ① 전체 수의 개수 확인: 16개
       ② 중간 값 찾기: 8번째 값=77
       ③ 96과 비교: 77보다 크므로 77 이후의 8개의 수에서 확인
       ④ 79-99까지의 8개의 값의 중간인 4번째 값 90과 비교
       ⑤ 90보다 큰 4개의 값의 중간인 2번째 값 96과 비교: 성공

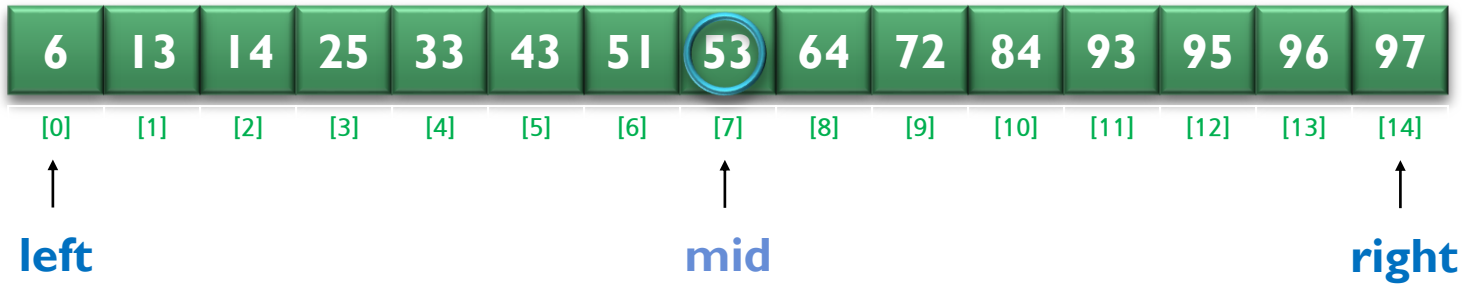


분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

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

분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

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



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

분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

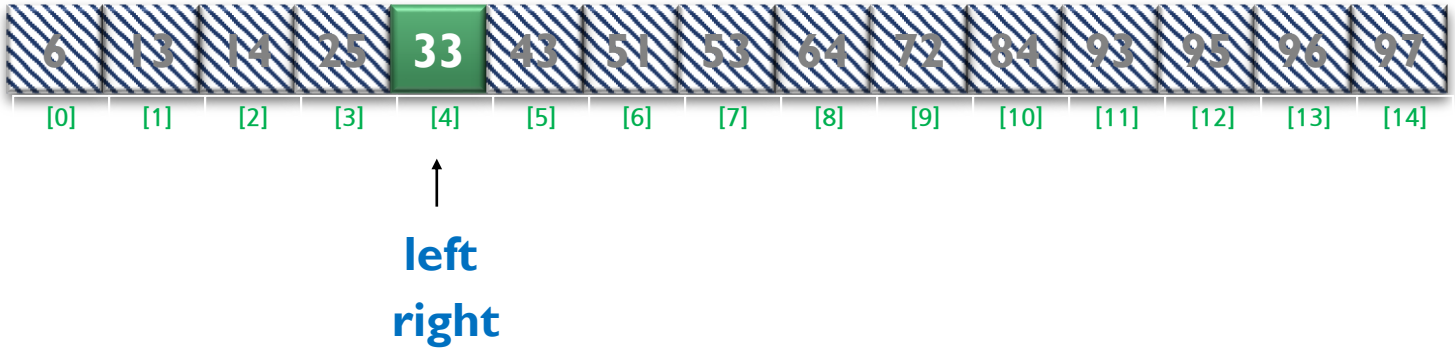
- 예] 33을 찾기 위한 이진 탐색
  - 반복하여 가운데 번지(0+6/2)에 해당하는 3번지를 mid로 지정

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

분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

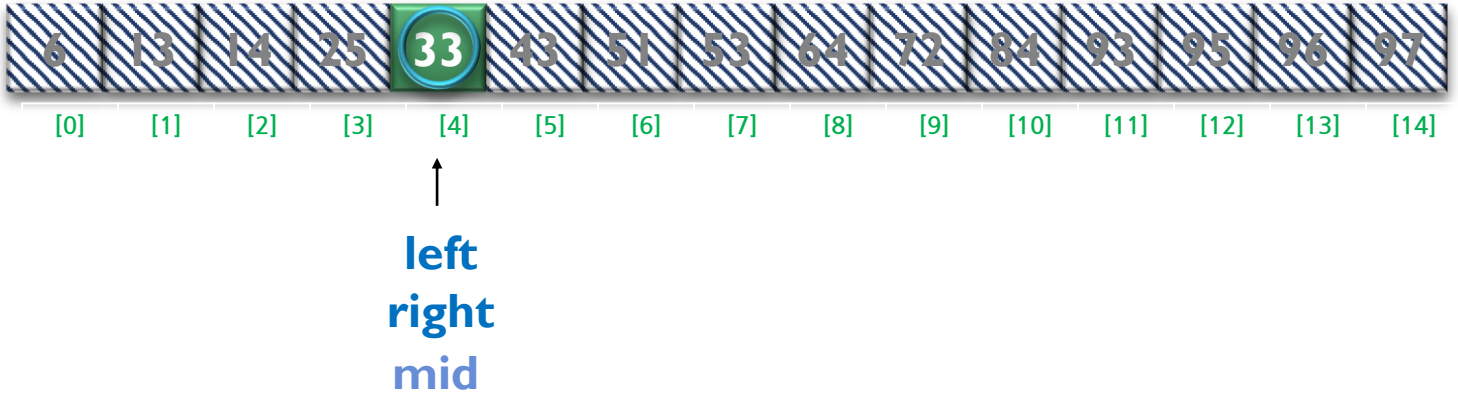
- 예] 33을 찾기 위한 이진 탐색
  - 반복하여 가운데 번지((4+6)/2)에 해당하는 5번지를 mid로 지정

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

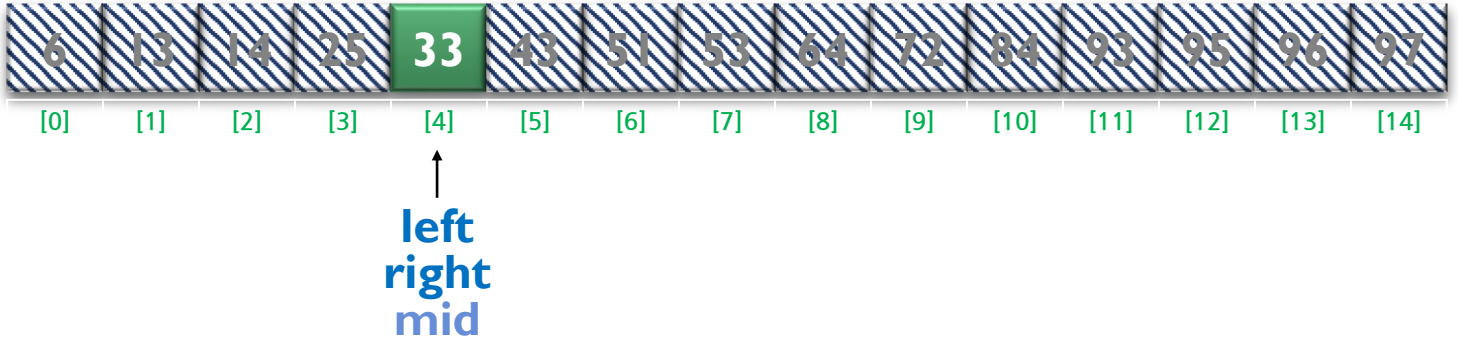


분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

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

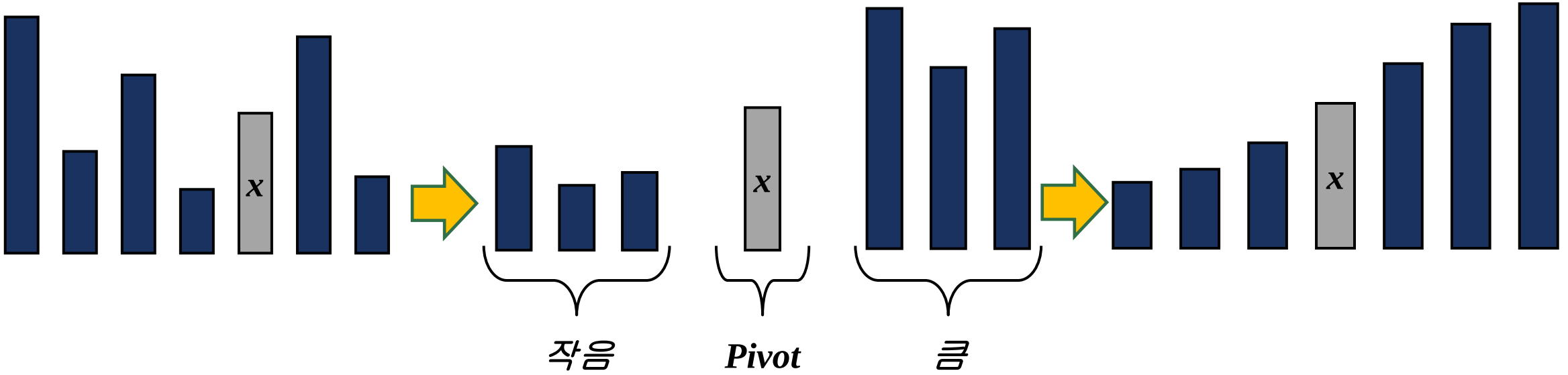


  - 모두가 모인 주소지가 찾고자 하는 값이므로 찾는 인덱스의 값은 4



분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - Pivot이라 하는 기준 값을 정한 후 분할 정복을 실행하는 방식
    - 기준키를 기준으로 작거나 같은 값을 지닌 데이터는 앞으로
    - 큰 값을 지닌 데이터는 뒤로 가도록 하여
  - 작은 값을 갖는 데이터와 큰 값을 갖는 데이터로 분리해가며 정렬하는 방법



분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - 임의의 값 5을 Pivot으로 선택

5	7	9	0	3	1	6	2	4	8

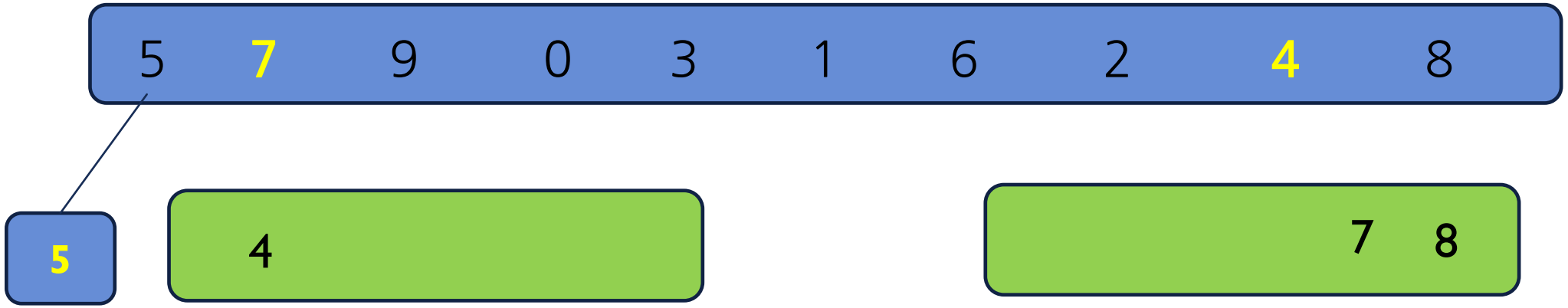
    - Pivot 의 값인 5를 기준으로 오른쪽으로 가면서 5 보다 큰 데이터를 선택하므로 '7'

5	**7**	**9**	0	3	1	6	2	4	8

5

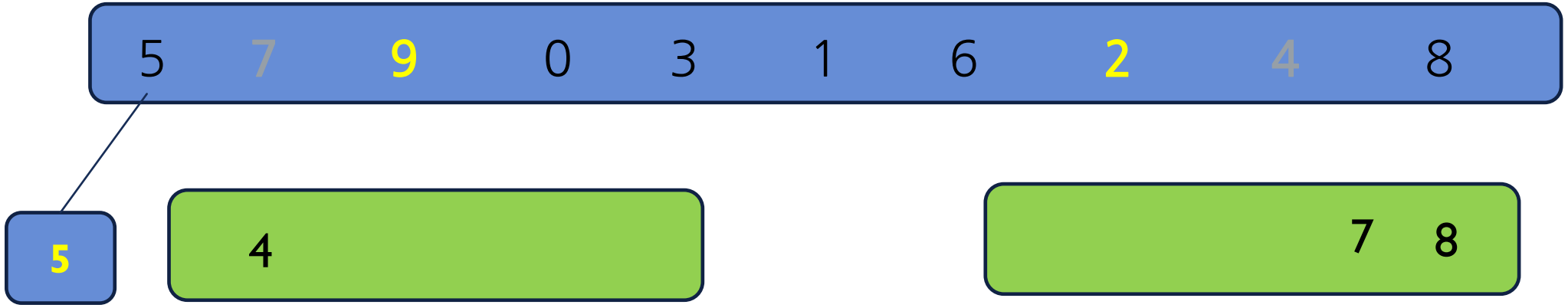
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - Pivot 의 값인 5를 기준으로 오른쪽에서 왼쪽으로가면서 5 보다 작은 데이터를 선택하므로 '4'
    - 두수의 위치를 바꿈



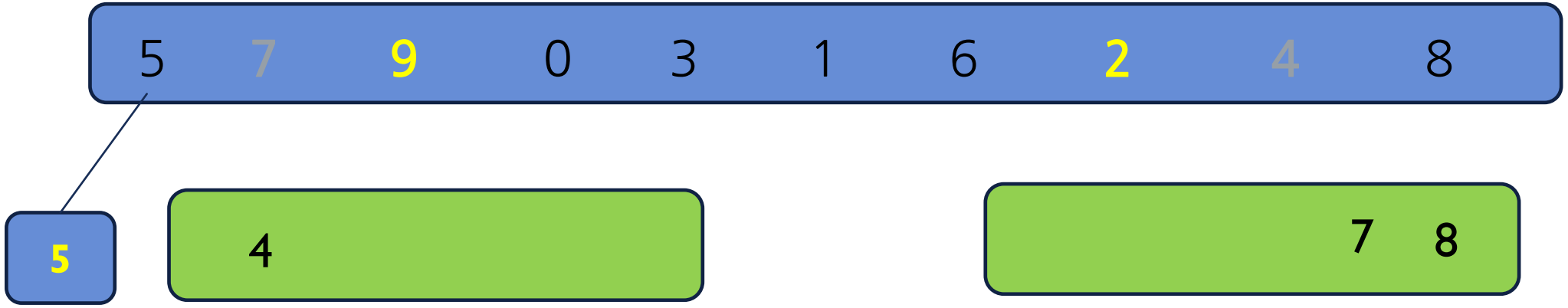
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - Pivot 의 값인 5를 기준으로 오른쪽으로 가면서 5 보다 큰 데이터 선택 '9'



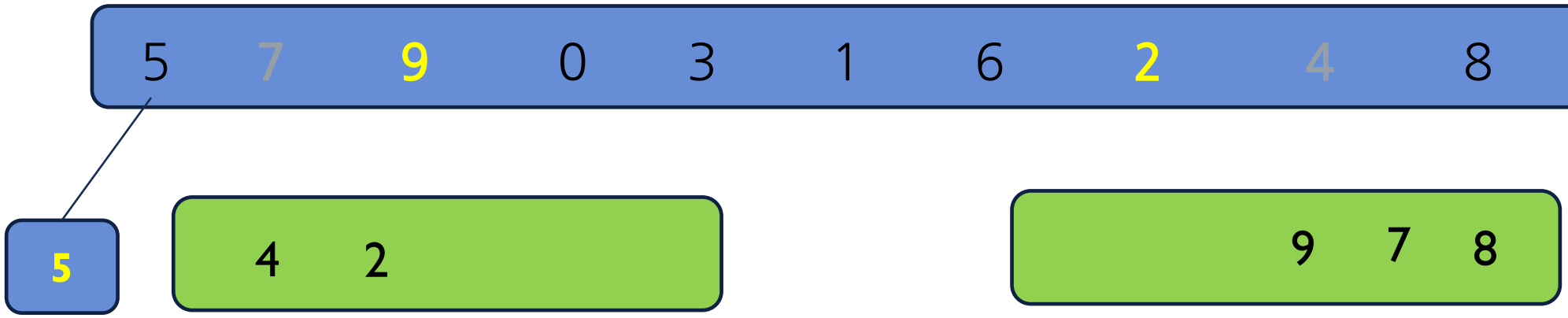
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - Pivot 의 값인 5를 기준으로 오른쪽에서 왼쪽으로 가면서 5 보다 작은 데이터 선택 '2'



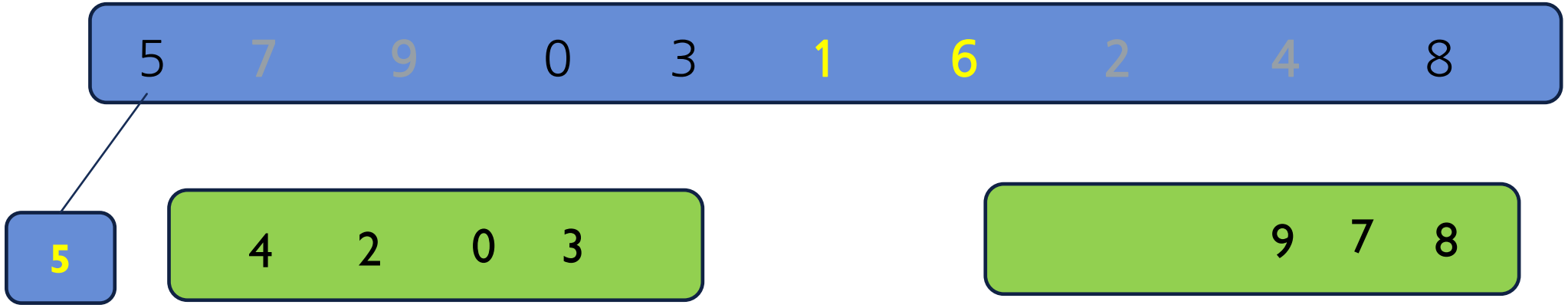
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - 두수를 바꿈



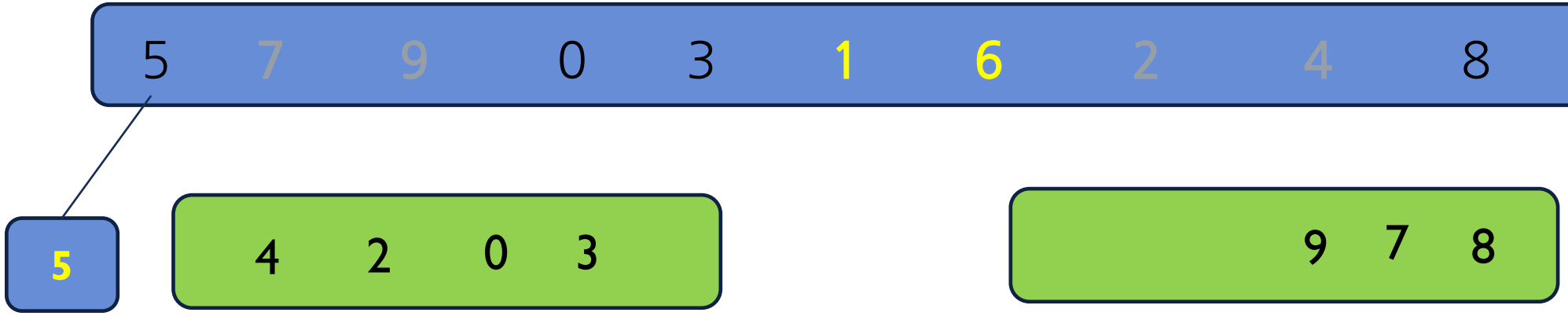
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - Pivot 의 값인 5를 기준으로 오른쪽으로 5 보다 큰 데이터를 선택하므로 '6'



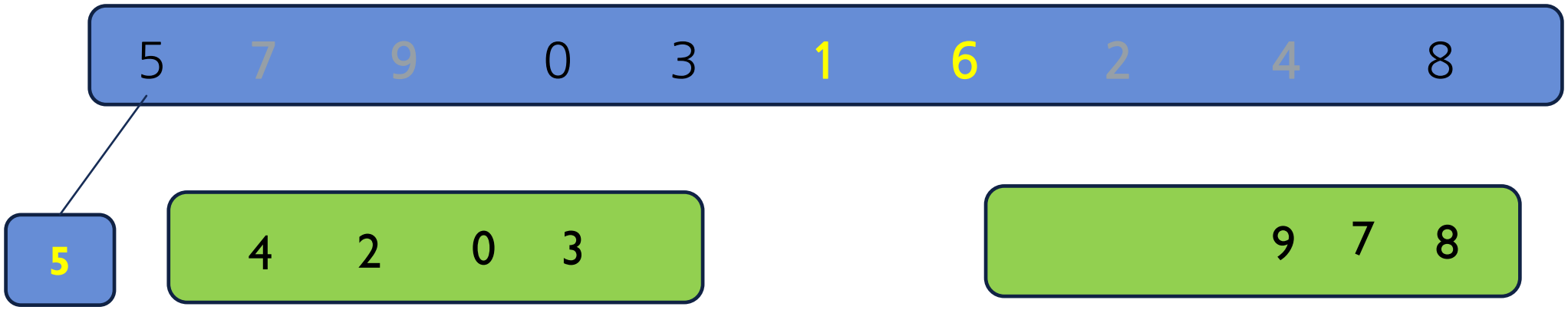
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - Pivot 의 값인 5를 기준으로 오른쪽에서 왼쪽으로 가면서 5 보다 작은 데이터를 선택하므로 '1'



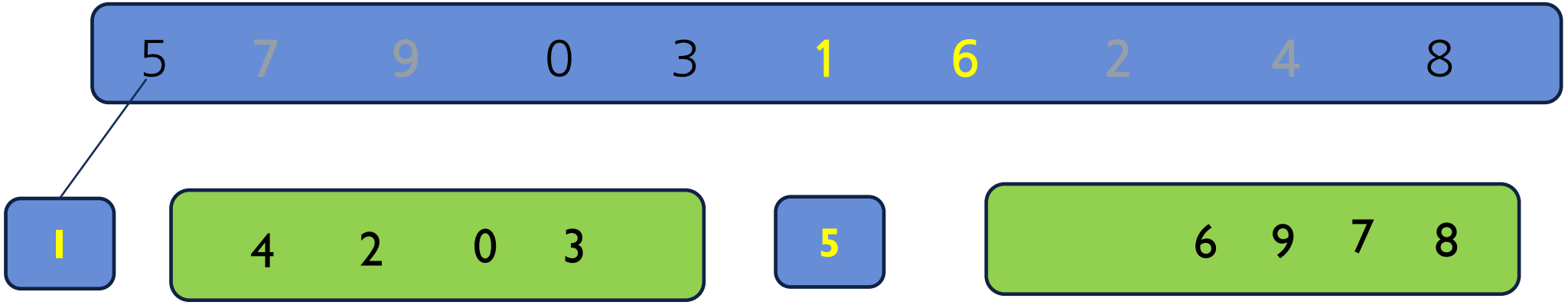
분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - 왼쪽방향과 오른쪽 방향이 서로 엇갈리게 되면 피벗과 작은 데이터의 위치를 변경



분할 정복 (DIVIDE-AND-CONQUER) 적용 문제

- 빠른 정렬 (Quicksort)
  - [5, 7, 9, 0, 3, 1, 6, 2, 4, 8]의 빠른 정렬
    - Pivot 의 값인 5를 기준으로 왼쪽은 작은 수가 오른쪽은 큰 수가 정렬됨



데이터 탐색의 이해

- 분할 정복 (Divide and Conquer) 알고리즘 개념
- 분할 정복 알고리즘의 예
- 분할 정복 적용 문제
  - 빠른정렬
  - 이진탐색