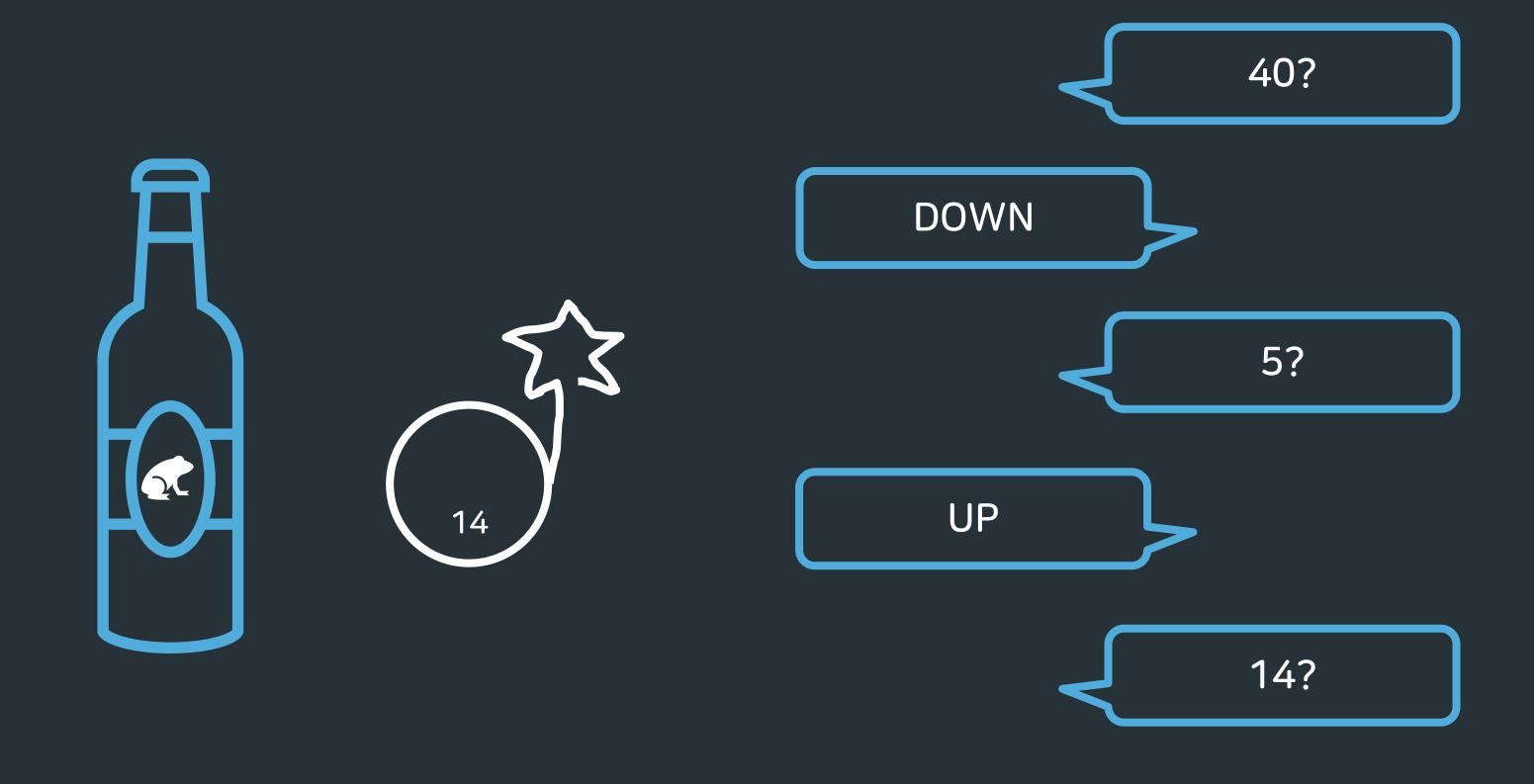




문제의 크기를 절반으로 줄이면서 빠르게 답을 찾는 알고리즘입니다. 코딩 테스트에선 주로 효율성을 보는 문제에 활용됩니다.

일상 속 이분 탐색





이런 문제가 있다고 해봅시다.

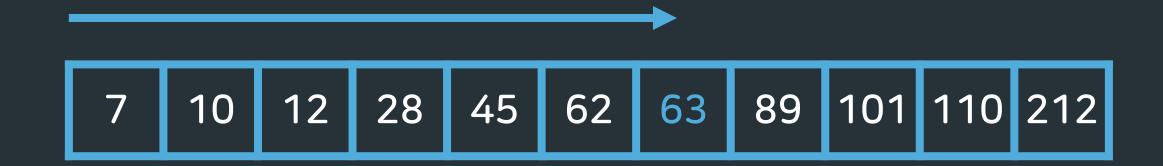




"이 안에 63이 있나?"

이런 문제가 있다고 해봅시다.





O(n)으로 찾을 순 있지만…

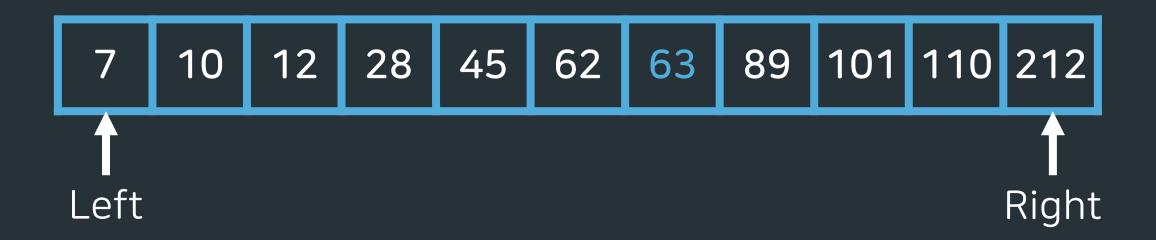
이분 탐색



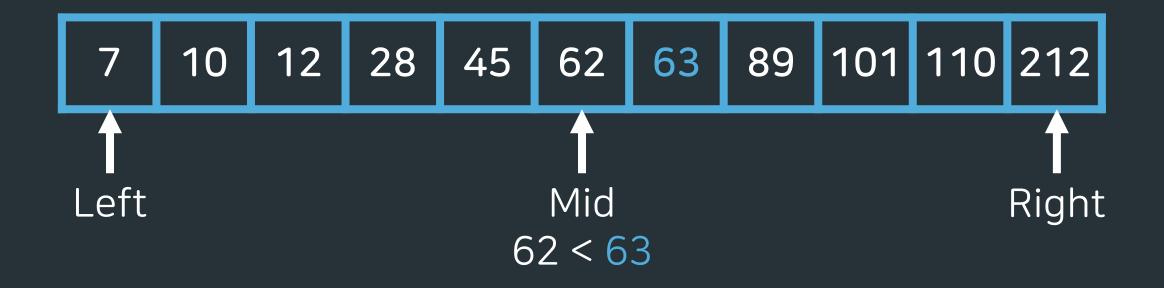
Binary Search

- 배열의 크기를 절반으로 줄이면서 답을 찾는 알고리즘
- 주로 반복문으로 구현
- 시간 복잡도는 O(logn)
- 입력 범위 N이 큰 편
- 알고리즘이 성립하기 위해선 배열이 반드시 정렬되어 있어야 함









● 연산 횟수: 1



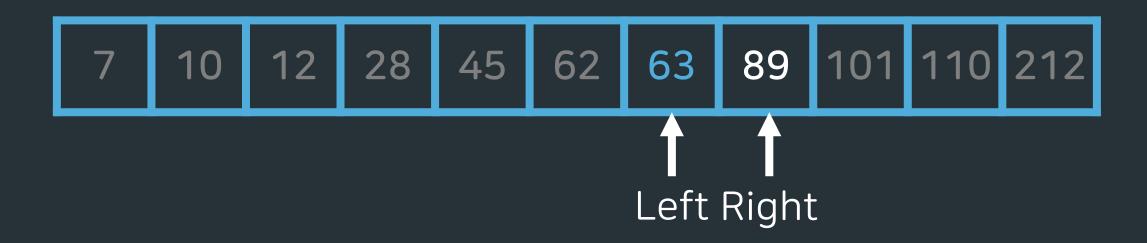




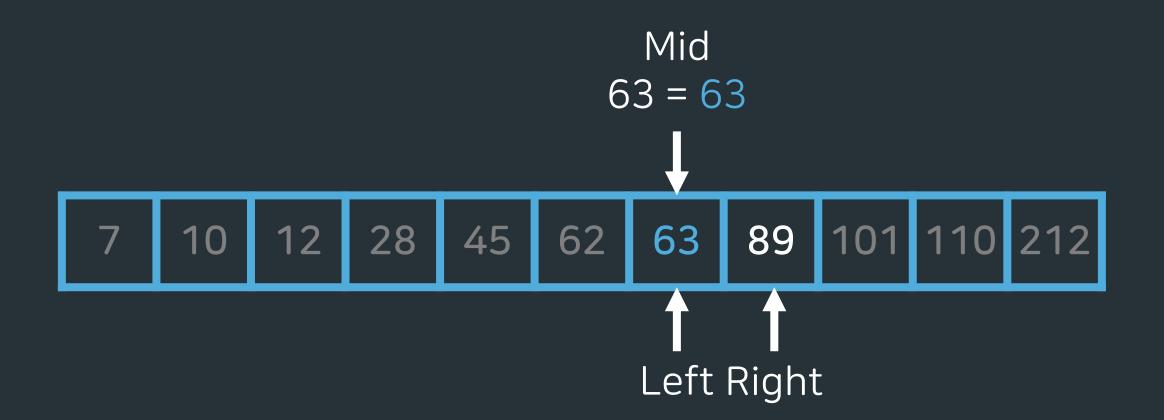


● 연산 횟수: 2





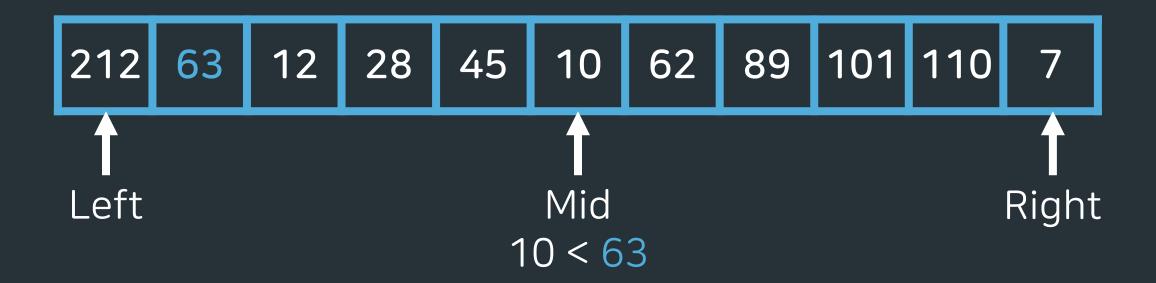




• 연산 횟수: 3

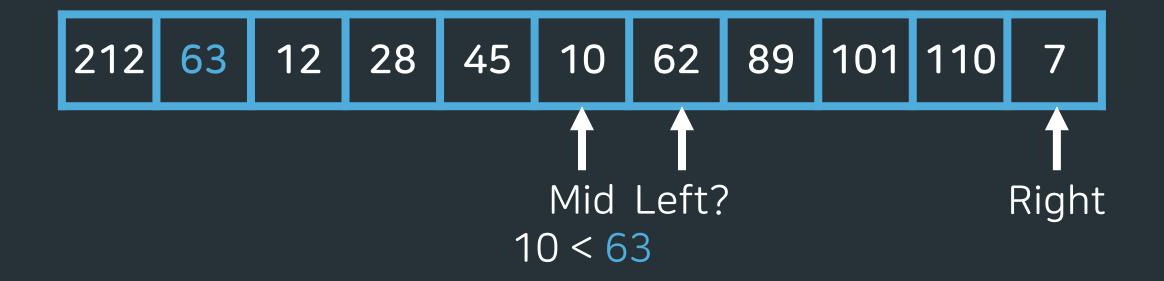
만약 정렬이 안 되어 있다면?





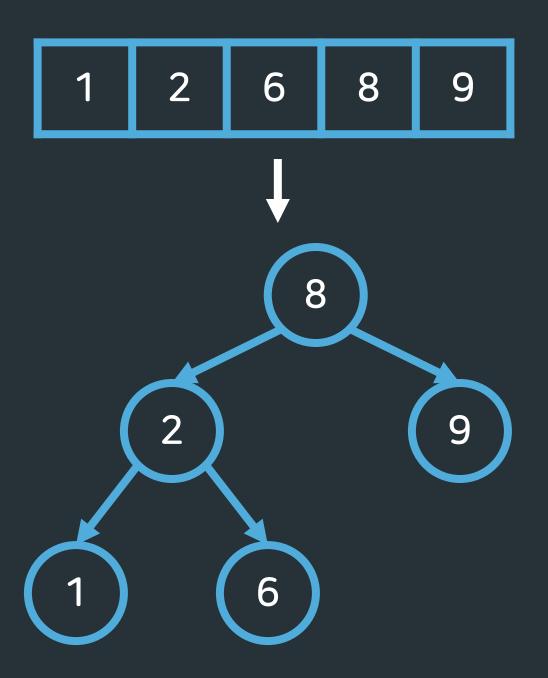
만약 정렬이 안 되어 있다면?





뭔가 낯설지 않은 이 기분





- 이분 탐색의 대상이 되는 원소들을 트리에 넣으면 BST!
- BST를 중위 순회(inorder)하면 배열이 나옴

기본 문제



/<> 1920번 : 수 찾기 - Silver 4

문제

● N개의 정수가 주어질 때, X라는 정수가 존재하는지 알아내라

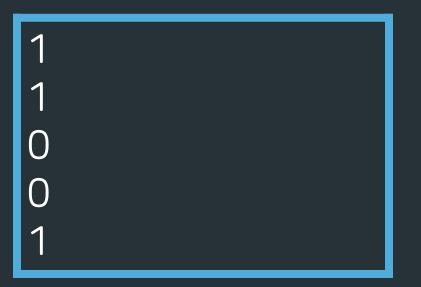
제한 사항

- N은 1 <= N <= 100,000
- 알아내야 하는 수 M의 개수는 1 <= M <= 100,000
- 입력되는 정수 k는 int 범위 내에 있음

예제 입력

5 41523 5 13795

계제 출력



기본 문제



/<> 1920번 : 수 찾기 - Silver 4

문제

● N개의 정수가 주어질 때, X라는 정<u>수가 존재하는지 알아</u>내라

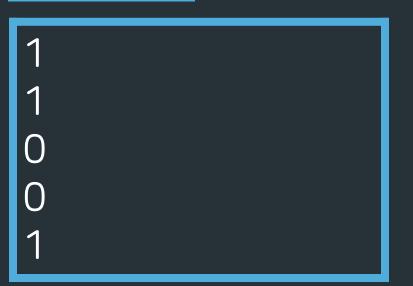
제한 사항

- N은 1 <= N <= 100,000
- 알아내야 하는 수 M의 개수는 1 <= M <= 100,000
- 입력되는 정수 k는 int 범위 내에 있음
- 1. 순차 탐색? -> 100,000 * 100,000 = 1e10(=100억) 시간초과
- 2. 셋? -> 100,000 * 100 + 100,000 * 100 = 2*1e7(=2000만) 메모리면에서 비효율적이나 가능 but 만약 정수의 위치를 요구하면 불가능한 방법

예제 입력

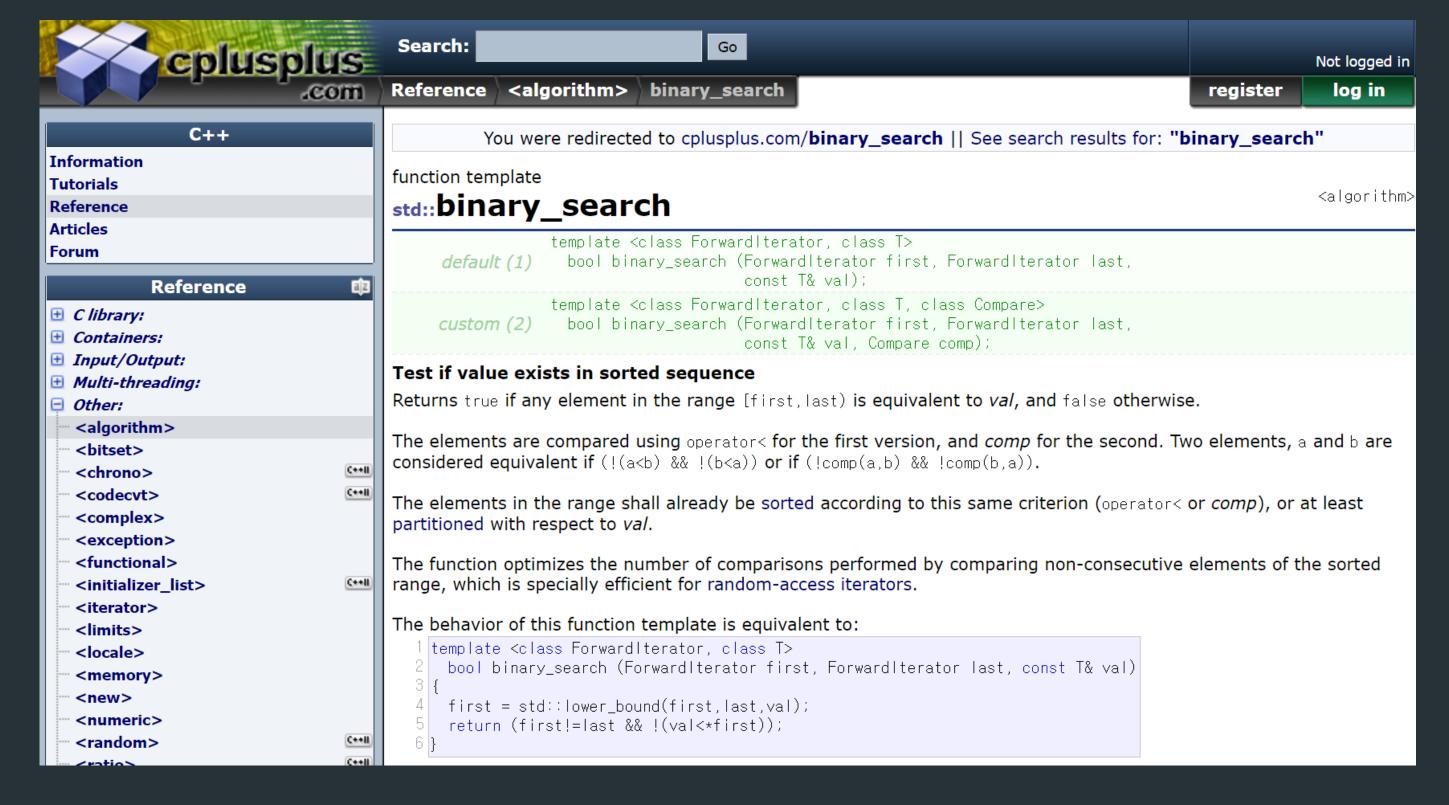
5 41523 5 13795

예제 출력



함수가 있긴 한데요…!





원소의 존재 여부만 리턴하는 간단한 함수이므로 구현하는 방법을 알아야 함

기본 문제



/<> 10816번 : 숫자 카드 2 - Silver 4

문제

- N개의 카드가 주어진다.
- M개의 정수에 대해 해당 숫자가 적힌 카드가 몇 장인가?

제한 사항

- N은 1 <= N <= 500,000
- 알아내야 하는 수 M의 개수는 1 <= M <= 500,000
- 입력되는 정수 k는 -10,000,000 <= k <= 10,000,000

예제 입력

10 6 3 2 10 10 10 -10 -10 7 3 8 10 9 -5 2 3 4 5 -10

예제 출력

30012002

기본 문제



/<> 10816번 : 숫자 카드 2 - Silver 4

문제

- N개의 카드가 주어진다.
- M개의 정수에 대해 해당 숫자가 적힌 카드가 몇 장인가?

제한 사항

- N은 1 <= N <= 500,000
- 알아내야 하는 수 M의 개수는 1 <= M <= 500,000
- 입력되는 정수 k는 -10,000,000 <= k <= 10,000,000

예제 입력

10 6 3 2 10 10 10 -10 -10 7 3 8 10 9 -5 2 3 4 5 -10

예제 출력

30012002

- 1. 순차 탐색? -> 500,000 * 500,000 = 25*1e10(=2500억) 시간초과
- 2. 셋? -> 500,000 * 700 + 500,000 * 700 = 7*1e8(=7억) 메모리면에서 비효율적이나 가능 but 만약 배열이 정렬됐을 때, 정수의 범위 인덱스를 요구하면 불가능한 방법

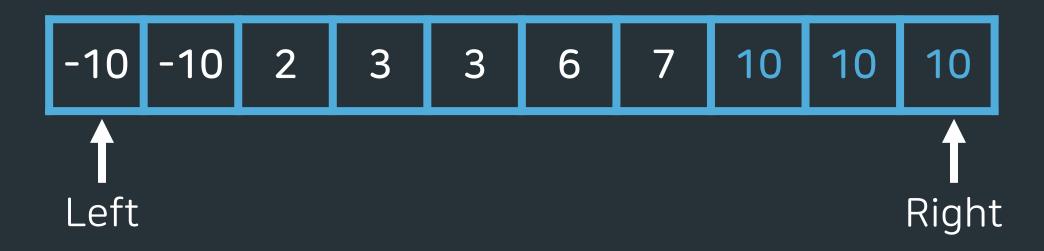
범위는 어떻게 찾지?





처음으로 10이 나오는 위치와 마지막으로 10이 나오는 위치?

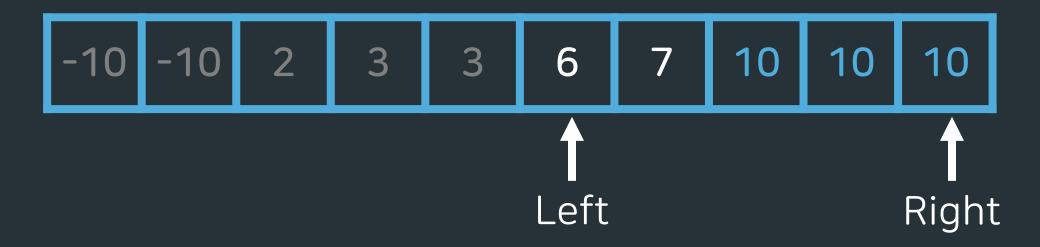








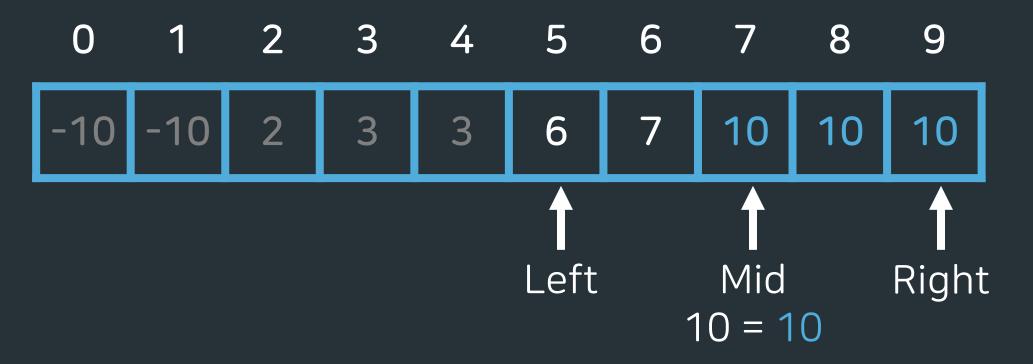






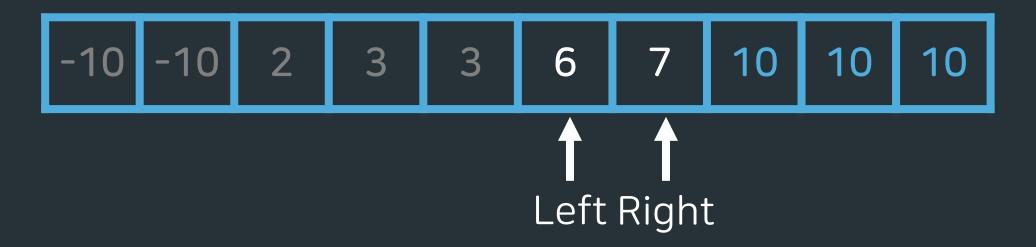




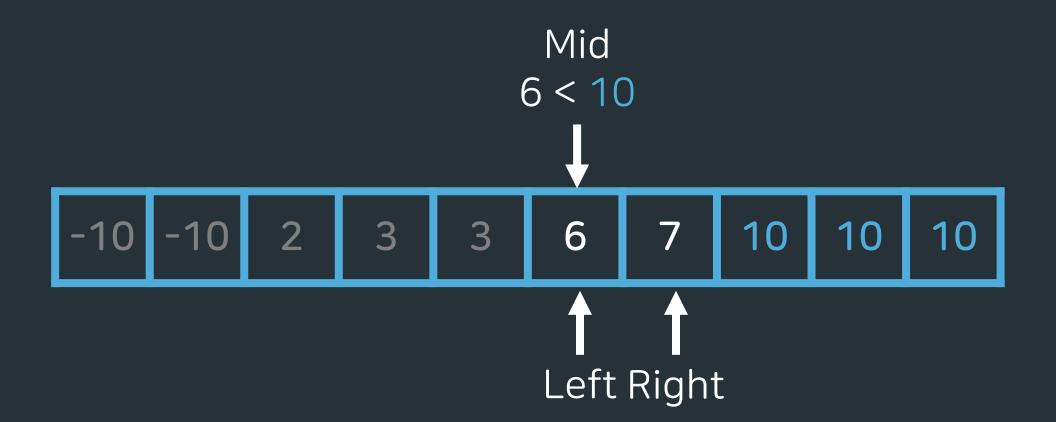


일단 7 위치에 있는데, 혹시 왼쪽에 10이 더 있지 않을까?





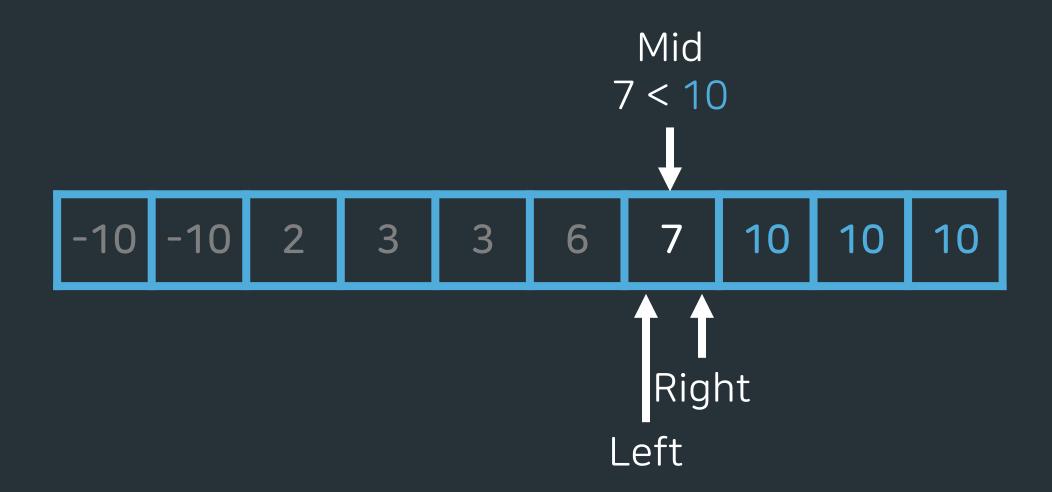








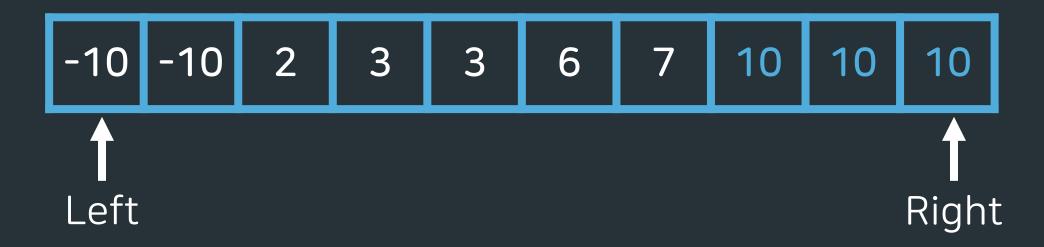












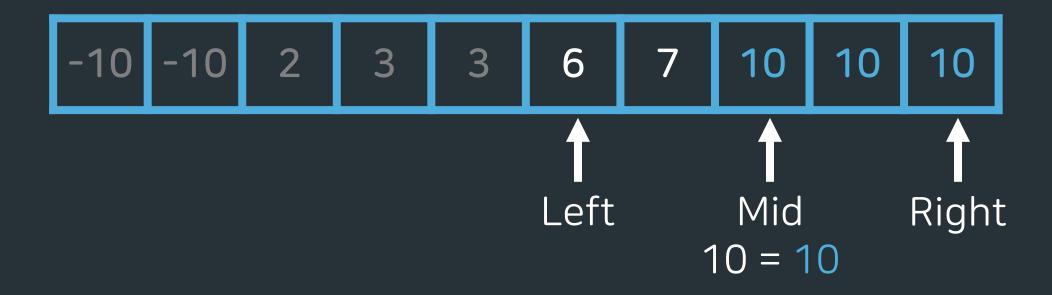




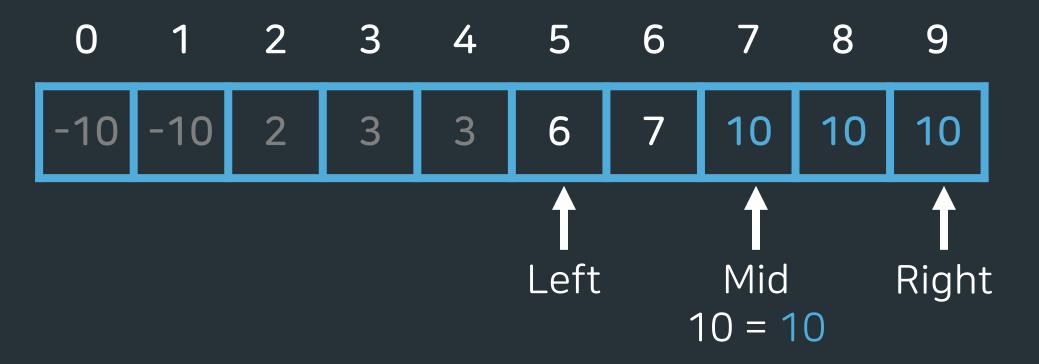






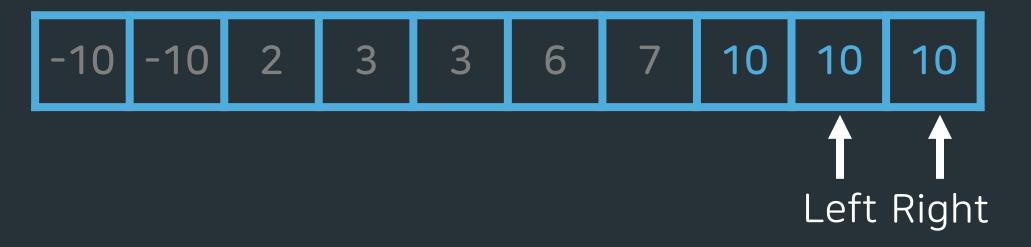




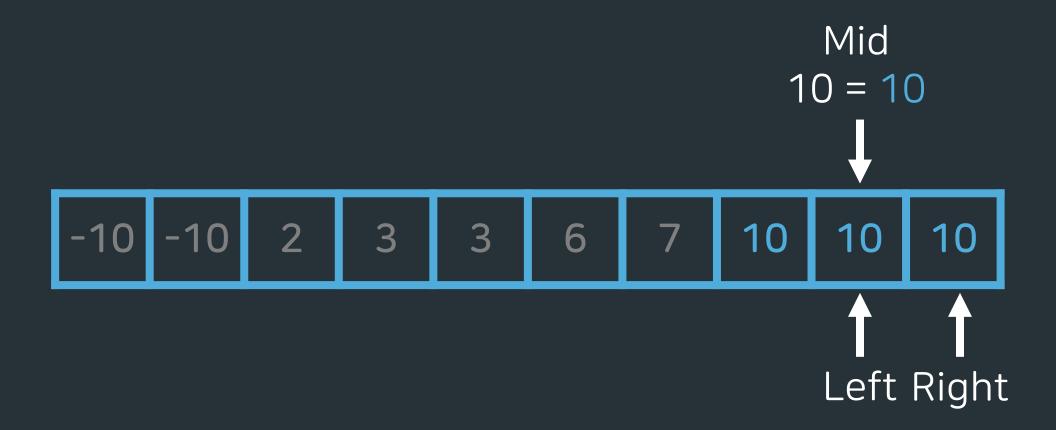


일단 7 위치에 있는데, 혹시 오른쪽에 10이 더 있지 않을까?

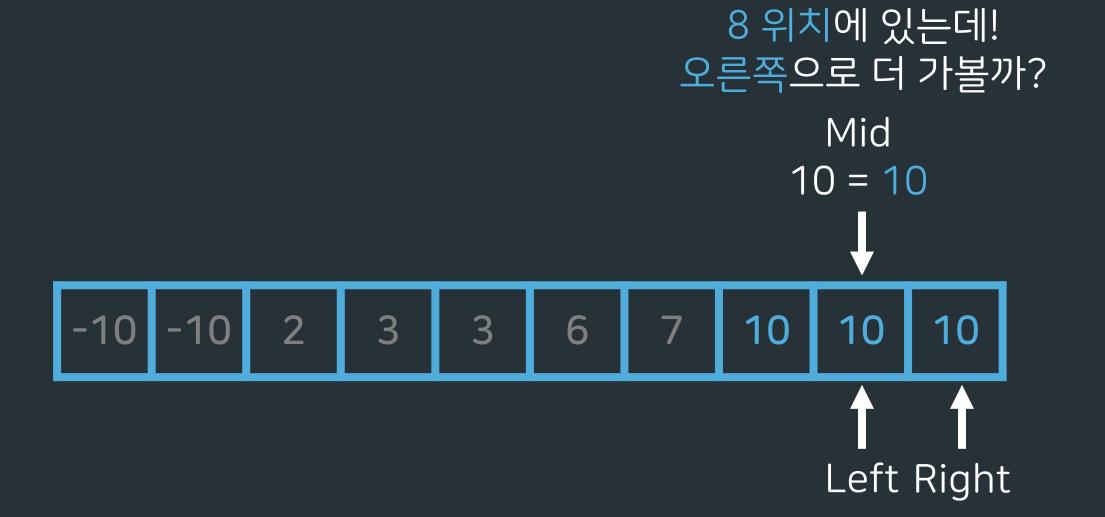








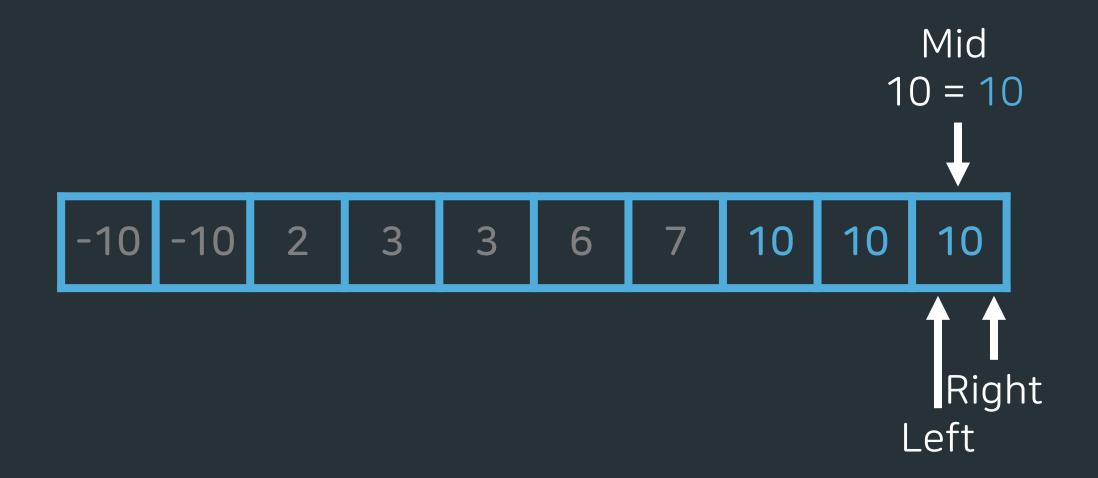




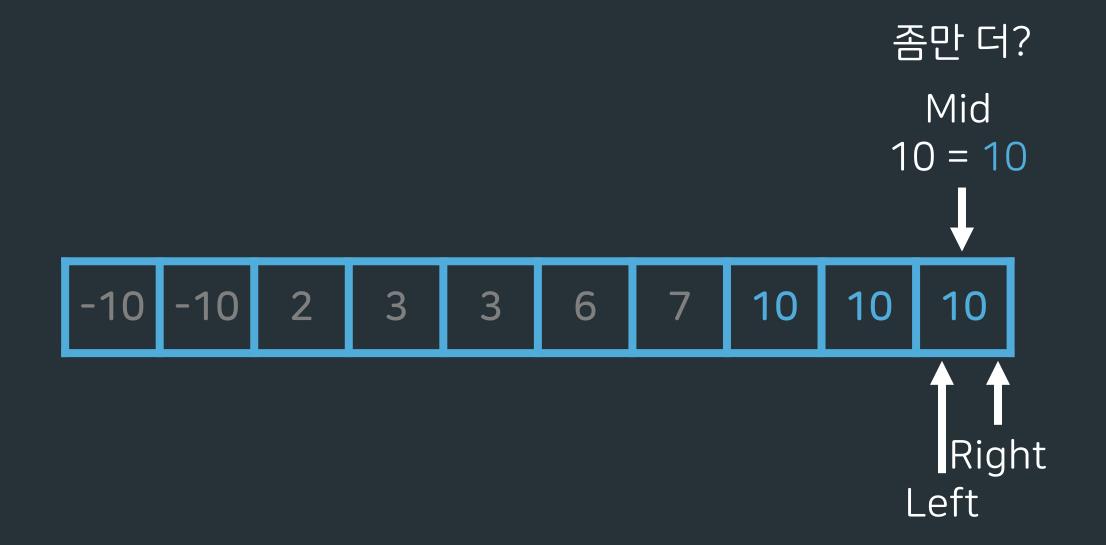




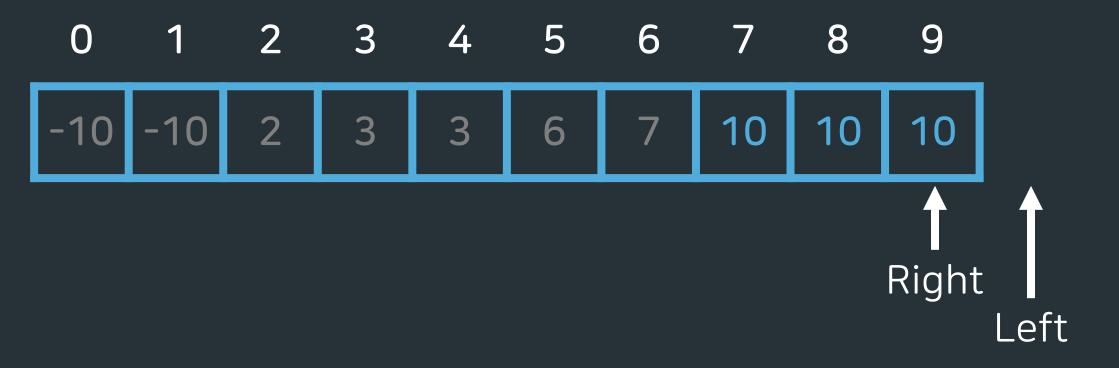




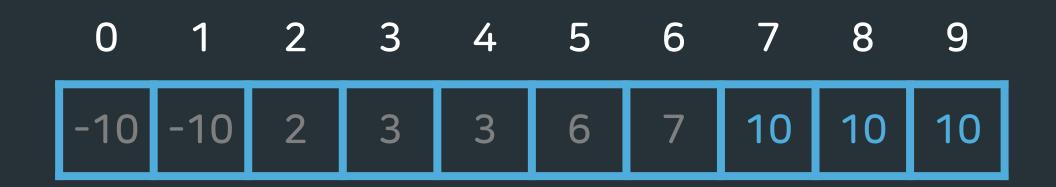












처음으로 10이 나오는 위치는 7 마지막으로 10이 나오는 위치는 9 10의 개수는 (9-7)+1 = 3

하한과 상한



Lower Bound

● 찾고자 하는 값인 X 이상의 수가 처음으로 나오는 위치

Upper Bound

● 찾고자 하는 값인 X를 초과하는 수가 처음으로 나오는 위치

하한과 상한



Lower Bound

● 찾고자 하는 값인 X 이상의 수가 처음으로 나오는 위치

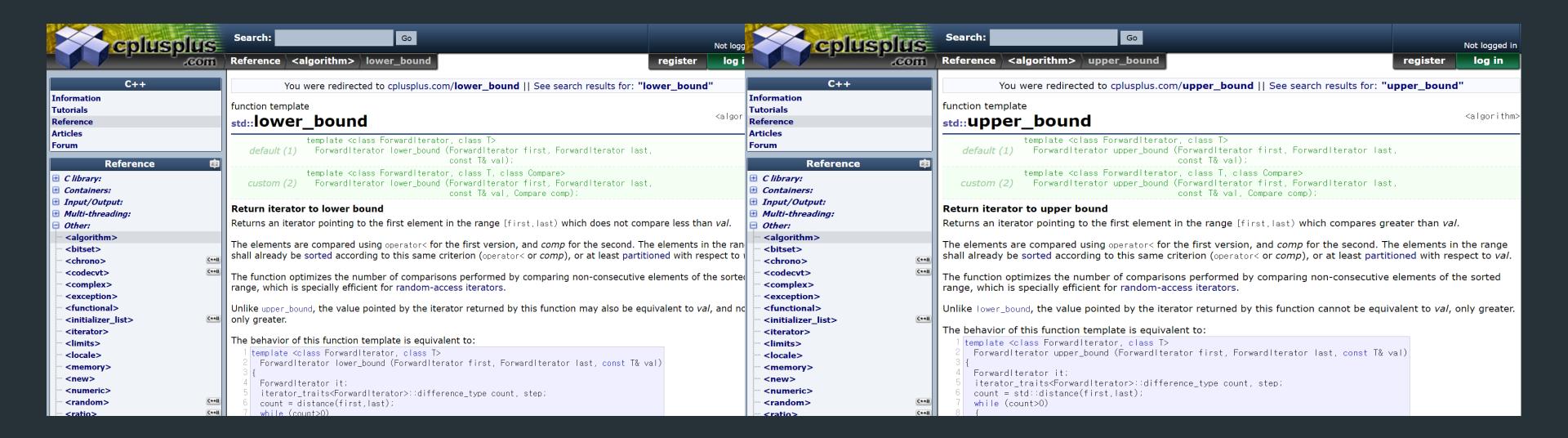
Upper Bound

찾고자 하는 값인 X를 초과하는 수가 처음으로 나오는 위치

* 정확히 말하면 10의 lower bound는 7이지만, 10의 upper bound는 10

함수가 있긴 한데요…!





이분 탐색 문제의 특징 때문에 구현하는 방법을 알아야 함

응용 문제



2110번 : 공유기 설치 - Silver 1

문제

- N개의 집이 수직선상에 놓여있다.
- 공유기는 C개가 있고, 한 집에 최대 하나만 설치할 수 있다.
- 공유기 C개를 설치할 때, 가장 인접한 두 공유기 사이의 최대 거리는?

제한 사항

- N의 범위는 2 <= N <= 200,000
- C의 범위는 2 <= C <= N
- 집의 좌표 x의 범위는 0 <= x <= 1,000,000,000

예제 입력

53

1

2

8

4

9

예제 출력

3

Parametric Search



Parametric Search

- 최적화 문제를 결정 문제로 바꾸는 알고리즘
- 가능한 답의 나열이 비내림차순 또는 비오름차순이어야 함
- 최댓값, 최솟값을 요구하는 문제라면 Parametric Search가 아닌지 의심!
- 탐색을 시작할 left, right를 잘 정해야 함

최적화 문제를 결정 문제로?



공유기 C개를 설치할 때, 가장 인접한 두 공유기 사이의 최대 거리는?



공유기 C개를 설치할 때, 가장 인접한 두 공유기 사이의 최대 거리는? -> 가장 인접한 두 공유기 사이의 거리가 K일 때, 공유기 C개를 설치할 수 있는가?

> 질문에 대한 답이 하나로 나오는 결정 문제로 바꾸기 가능한 모든 K 중 최<u>댓값이 두 공유기 사이의 최대거리</u>

가능한 답의 나열





5원짜리 동전 3개와 5원 이하의 동전들로 나타낼 수 있는 최댓값은? *단, 남은 금액을 5로 나눈 값이 1이상이라면 반드시 5원을 사용해야 한다.

불가능한 답의 나열



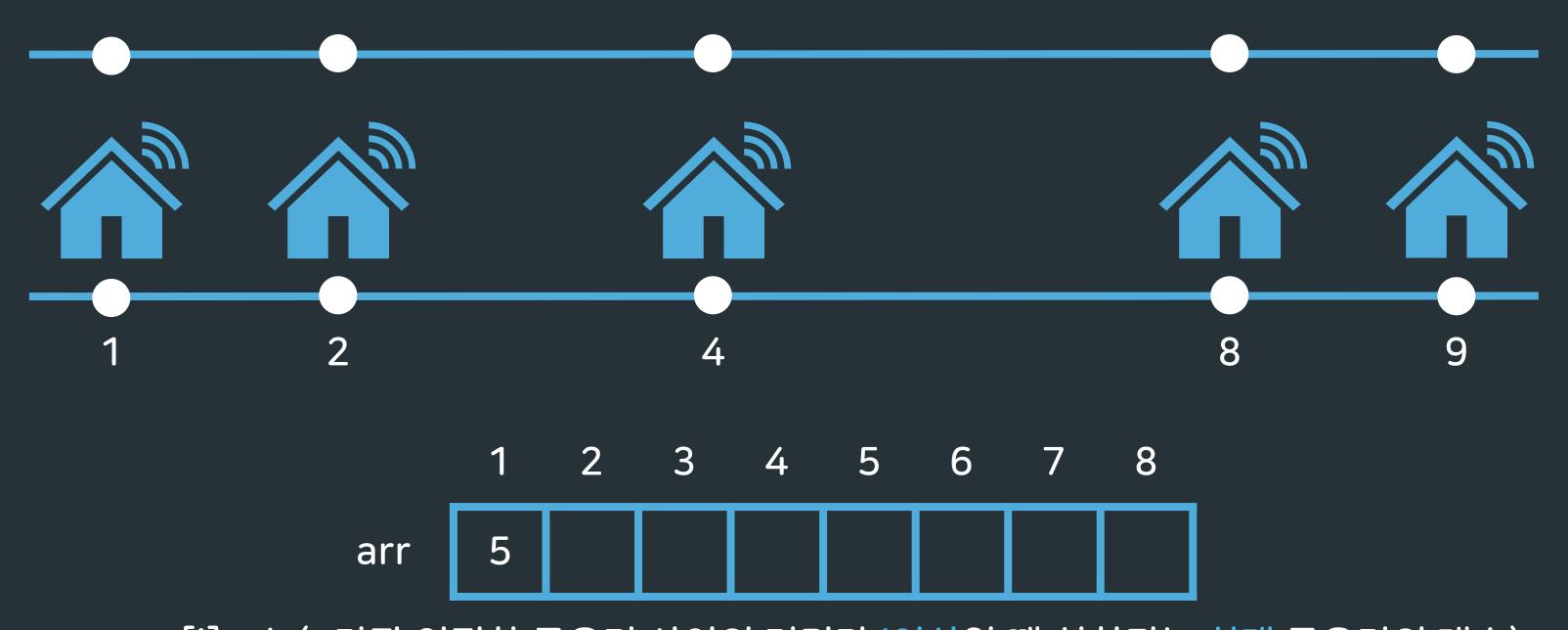


5원짜리 동전과 1원짜리 동전 총 4개를 사용하여 나타낼 수 있는 최댓값은? *단, 남은 금액을 5로 나눈 값이 1이상이라면 반드시 5원을 사용해야 한다.

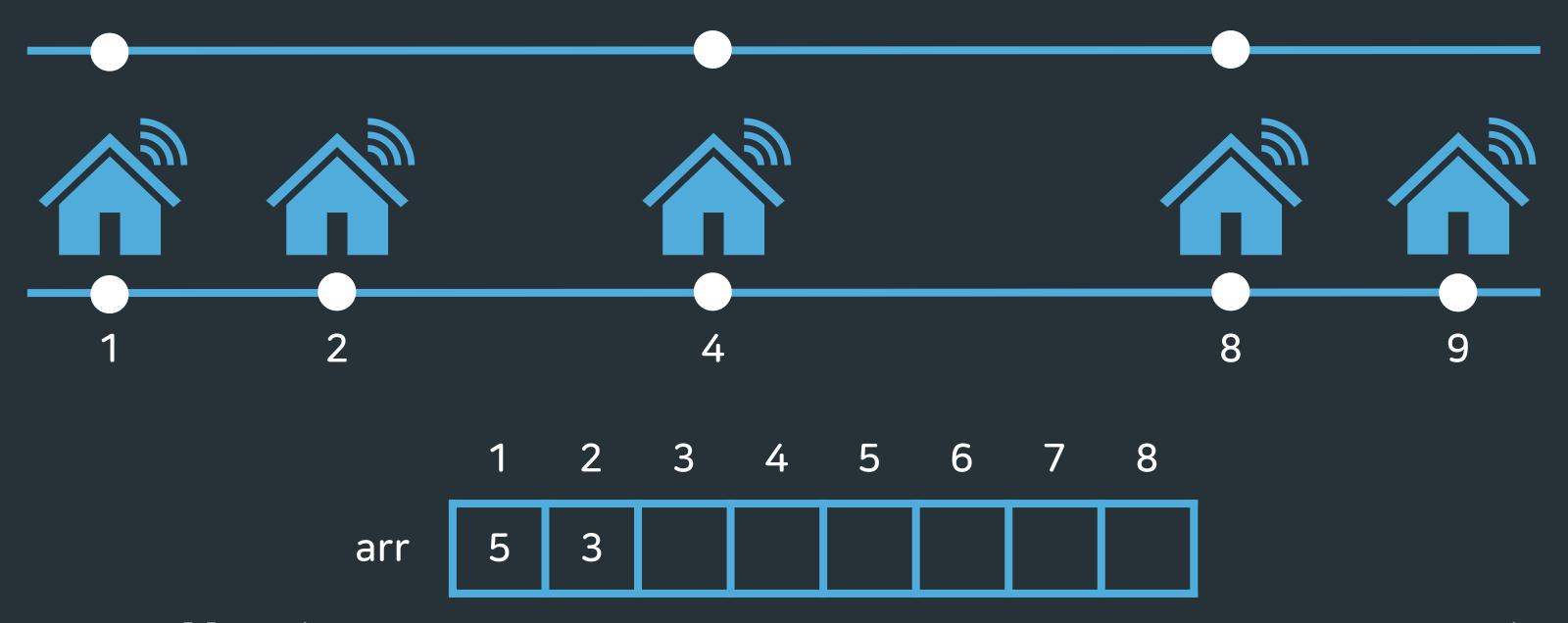




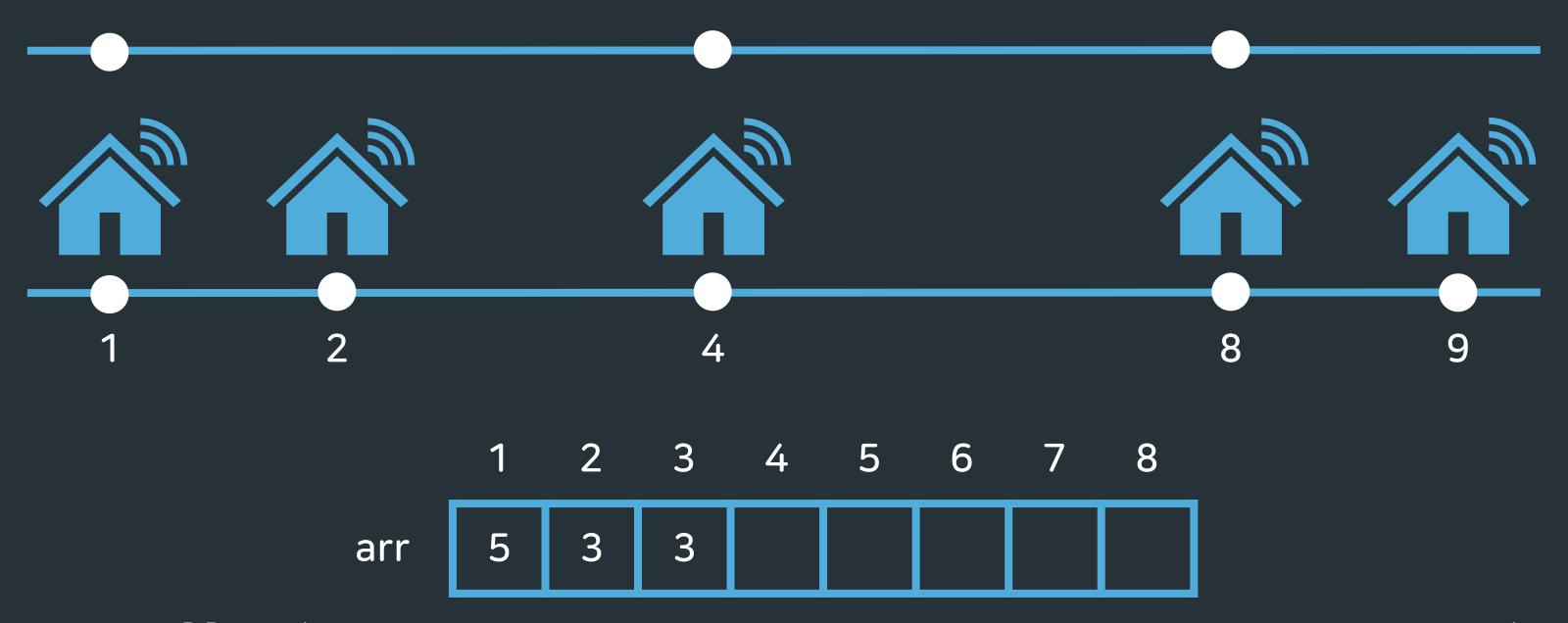




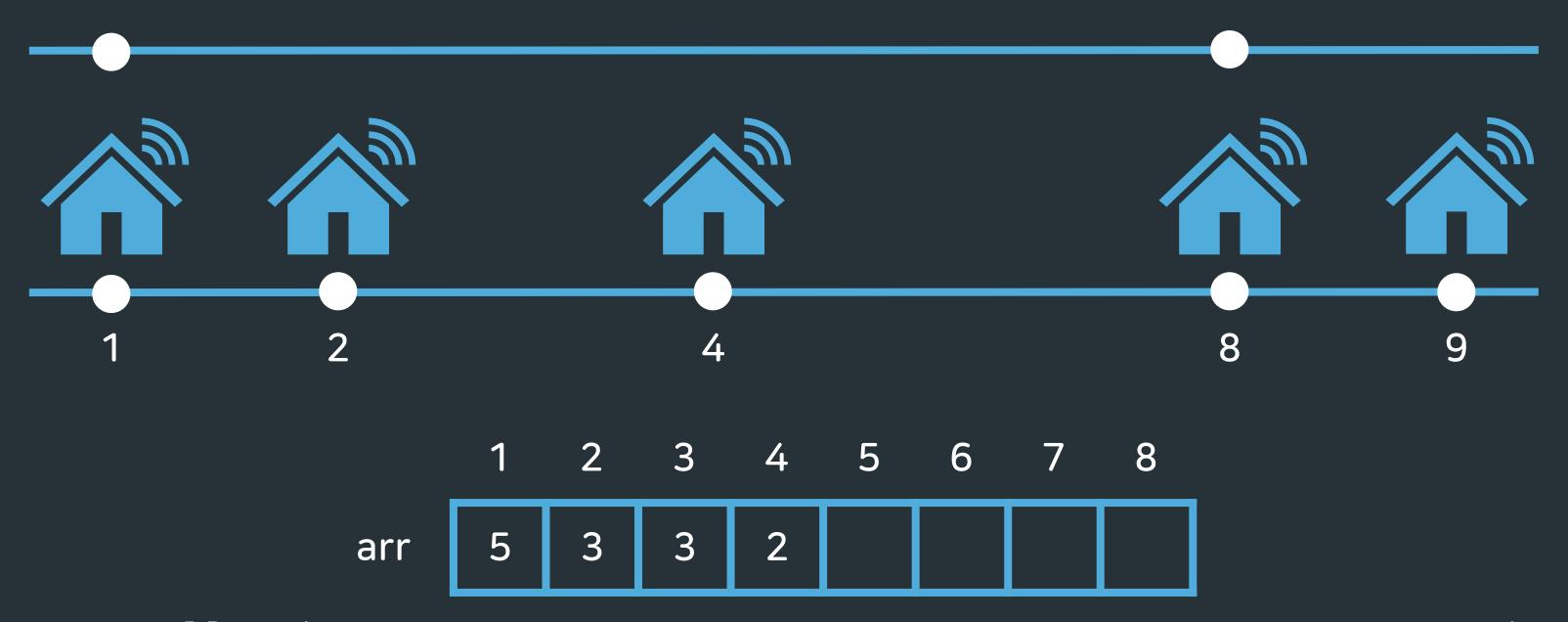




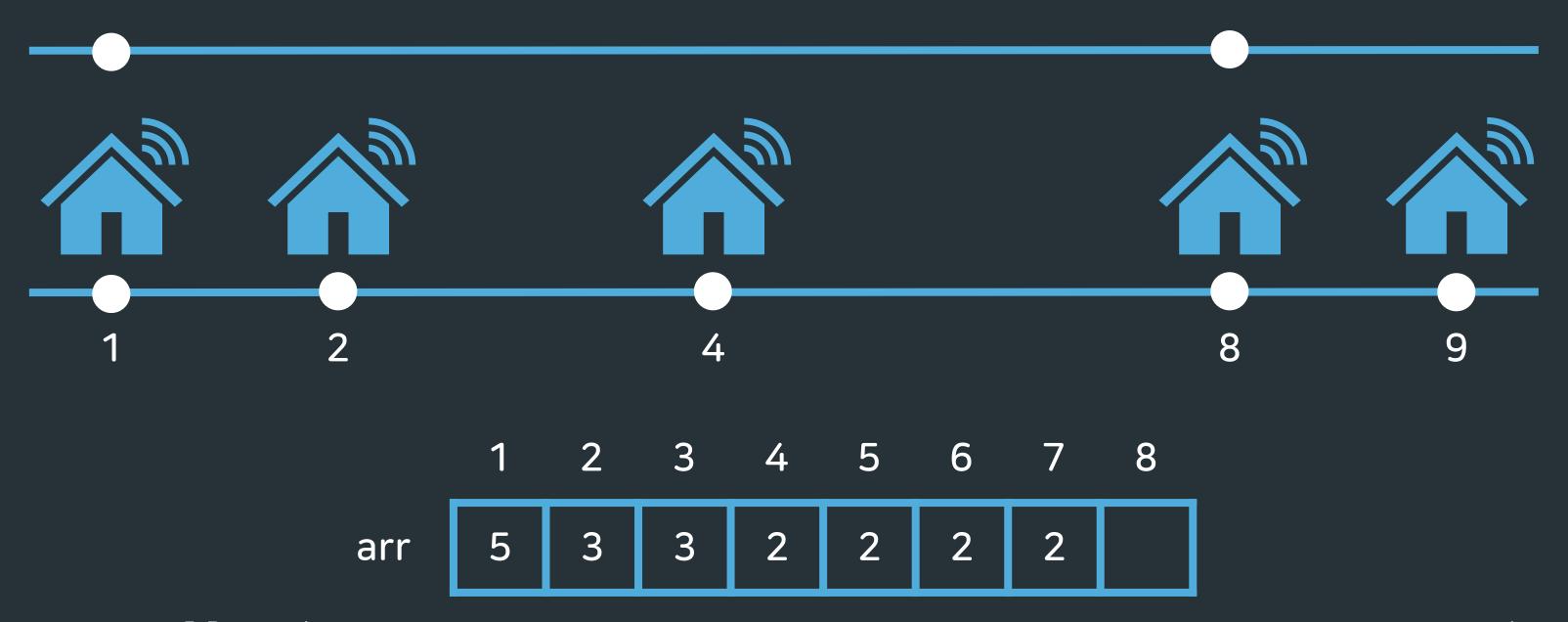




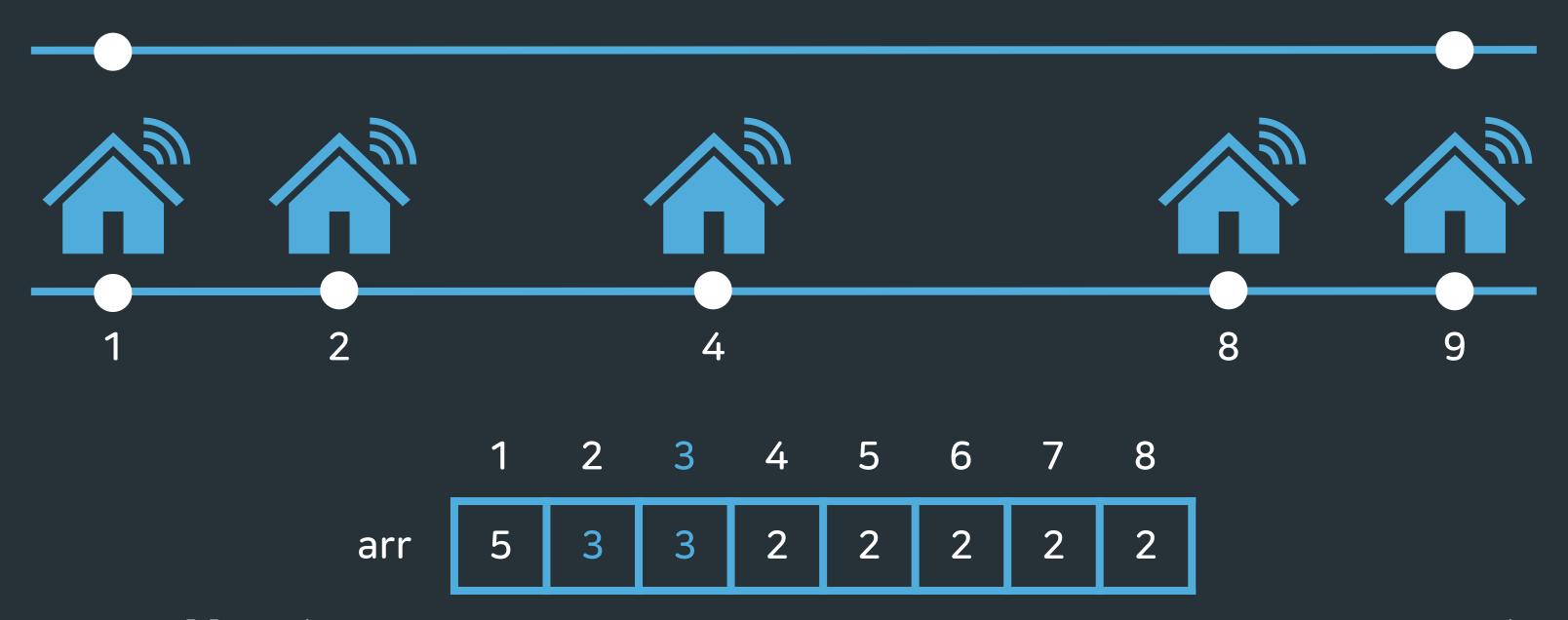












응용 문제



2110번 : 공유기 설치 - Silver 1

문제

- N개의 집이 수직선상에 놓여있다.
- 공유기는 C개가 있고, 한 집에 최대 하나만 설치할 수 있다.
- 공유기 C개를 설치할 때, 가장 인접한 두 공유기 사이의 최대 거리는?

제한 사항

- N의 범위는 2 <= N <= 200,000
- C의 범위는 2 <= C <= N
- 집의 좌표 x의 범위는 0 <= x <= 1,000,000,000

예제 입력

53

1

2

2

4

9

예제 출력

3

마무리



정리

- 배열의 원소가 정렬된 상태라면 이분 탐색보다 더 빨리 특정 원소를 찾을 수는 없음! O(logn)
- 분할 정복과 달리 문제를 나눈 뒤, 답이 없을 문제의 절반을 버림
- 코딩 테스트에는 주로 Parametric Search 문제로 등장
- 최댓값, 최솟값이라는 키워드가 보인다면 일단 이분 탐색을 의심해보자
- 효율성 테스트 문제로 아주 많이 출제됨

이것도 알아보세요

● (C++)원소를 정렬된 상태로 저장하는 맵과 셋에도 lower, upper bound가 있습니다. 한 번 찾아보세요

과제



필수

- /<> 14500번 : 테트로미노 Gold 4
- /<> 10815번 : 숫자 카드 Silver 4
- /<> 16401번 : 과자 나눠주기 Silver 2

도전

- /<> 2343번 : 기타 레슨 Silver 1
- /<> 3079번 : 입국 심사 Gold 5

과제 마감일



과제제출 마감

~ 5월 9일 화요일 18:59

추가제출 마감

~ 5월 12일 목요일 23:59