

이진탐색 알고리즘

→ 순차탐색: 리스트 안에 있는 특정한 데이터를 찾기 위해
앞에서부터 데이터를 하나씩 확인 하는 방법

→ 이진탐색: 정렬되어 있는 리스트에서 탐색 범위를 절반씩
줄여가며 데이터를 탐색하는 방법

→ 이진 탐색은 시작점, 끝점, 중간점을 이용하여
탐색 범위를 설정한다.

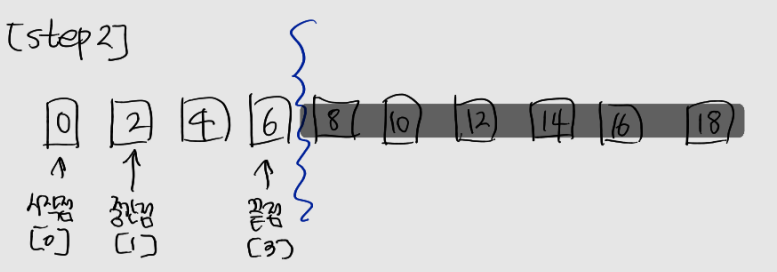
이진탐색 동작예시

• 이미 정렬된 10개의 데이터 중에서 값이 4인 원소 찾기.

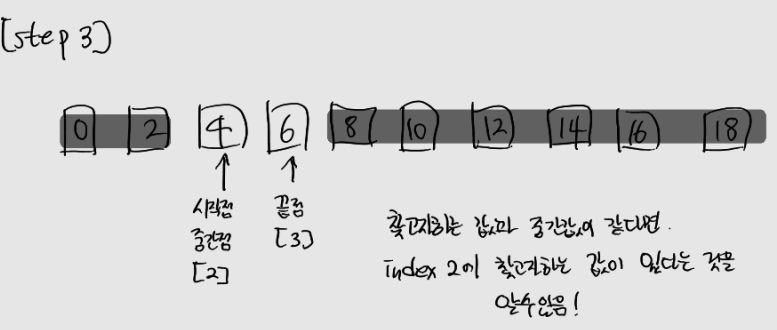


[step 1] 시작점 = 0, 끝점 = 9, 중간점 = 4 (소속점 이하 세기)
Index 설정.
↓
중간점이 두개 존재시에
소속점 제거해서 4

찾고자 하는 값보다 중간값이 크다면
끝점을 중간값의 왼쪽으로 이동.



찾고자 하는 값보다 중간값이 작다면
시작점을 중간점 오른쪽으로 이동



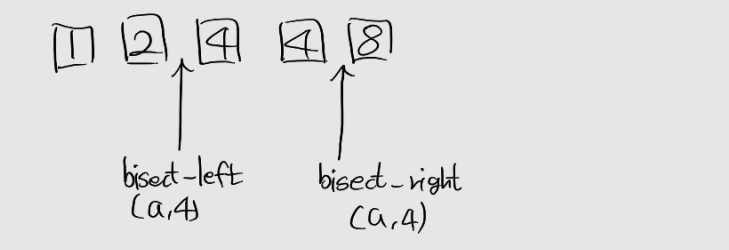
이진탐색의 시간 복잡도

- 단테마다 탐색 범위를 2로 나누는 것과 동일하므로
연산횟수는 $\log_2 N$ 이 되며
- 예를 들어 초기 데이터 개수가 32개 일때, 이상적으로 1단계를
거치면, 16개 이하의 데이터만 남는다
 - 2단계를 거치면 8개 이하 남는다
 - 3단계를 거치면 4개 이하 남는다
- 대다수 이진 탐색은 탐색 범위를 절반씩 줄이며,
시간 복잡도는 $O(\log N)$ 을 보장

→ python (이진탐색 - 재귀적 구현 - 예시코드.py)
→ python (이진탐색 - 반복문 구현 - 예시코드.py)

파이썬 이진 탐색 라이브러리.

- `bisect - left(a, x)`: 정렬된 순서를 유지하면서 배열 a에서
가장 작은 값보다 큰 원소를 반환
- `bisect - right(a, x)`: 정렬된 순서를 유지하면서 배열 a에서
가장 큰 값보다 작은 원소를 반환



→ python (이진탐색 - 라이브러리 - 예시.py)
↑
값이 특정 범위에 속하는 데이터 개수 구하기 포함.

Python Code

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1	2	3	3	3	3	4	4	8	9
left-value	4	right-value	4	right-index	8	left-index	6	result	2
-1	3	0	6	6					

파라메트릭 서치

→ 파라메트릭 서치란 최적화 문제를 찾는 문제 ("여" or "이니")로 바꾸어
검색하는 기법

→ 예시 : 특정한 조건을 만족하는 가장 알맞은 값을 빠르게 찾는 최적화 문제

→ 일반적으로 코딩에서 파라메트릭 서치 문제는 이진탐색을 이용하여 해결 가능

<문제> 떡볶이 떡 만들기 : 문제 설명

- 오늘 동빈이는 여행 가신 부모님을 대신해서 떡집 일을 하기로 했다. 오늘은 떡볶이 떡을 만드는 날이다. 동빈이네 떡볶이 떡은 재밌게도 떡볶이 떡의 길이가 일정하지 않다. 대신에 한 봉지 안에 들어가는 떡의 총 길이는 절단기로 맞춰준다.
- 절단기에 높이(h)를 지정하면 줄지어진 떡을 한 번에 절단한다. 높이가 h보다 긴 떡은 h 위의 부분이 잘릴 것이고, 낮은 떡은 잘리지 않는다.

EX)

- 높이가 19, 14, 10, 17cm인 떡이 나란히 있고 절단기 높이를 15cm로 지정하면 자른 뒤 떡의 높이는 15, 14, 10, 15cm가 될 것이다. 잘린 떡의 길이는 차례대로 4, 0, 0, 2cm이다. 손님은 6cm 만큼의 길이를 가져간다.
- 손님이 왔을 때 요청한 총 길이가 M일 때, 적어도 M 만큼의 떡을 얻기 위해 절단기에 설정할 수 있는 높이의 최댓값을 구하는 프로그램을 작성하시오.

<문제 조건>

[입력 조건]

- 첫째 줄에 떡의 개수 N과 요청한 떡의 길이 M이 주어진다. ($1 \leq N \leq 1,000,000$ & $1 \leq M \leq 2,000,000,000$)
- 둘째 줄에는 떡의 개별 높이가 주어진다. 떡 높이의 총합은 항상 M 이상이므로, 손님은 필요한 양만큼 떡을 사갈 수 있다. 높이는 10억보다 작거나 같은 양의 정수 또는 0이다.

[출력 조건]

- 적어도 M만큼의 떡을 집에 가져가기 위해 절단기에 설정할 수 있는 높이의 최댓값을 출력한다.

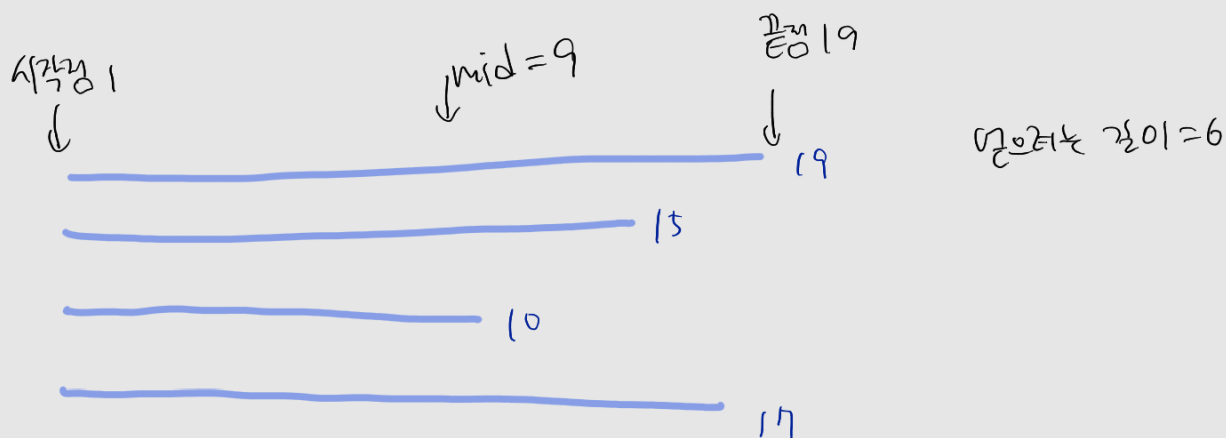
[입력 예시]

4 6
19 15 10 17

[출력 예시]

15

-> python(떡볶이 떡 만들기.py)



- ① mid에서 자른 길이가 원하는 길이보다 큰 경우 $start = mid + 1$
- ② mid에서 자른 길이가 원하는 길이보다 작은 경우 $end = mid - 1$
- ③ mid에서 자른 길이가 원하는 길이와 같은 경우 (mid)

Hand Code,,,

[Step] start mid end
1 9 19

- ① end-mid $\Rightarrow 10$
 - ② 15-9 $\Rightarrow 6$
 - ③ 10-9 $\Rightarrow 1$
 - ④ 17-9 $\Rightarrow 8$
- $\Rightarrow 25$ start \approx mid+1

start mid End
10 14 19

- ① end-mid = 5
 - ② 15-14 = 1
 - ③ 10-14 = \otimes
 - ④ 17-14 = 3
- $\Rightarrow 4$ start \approx mid+1

start mid End
15 17 19

- ① 19-17 = 2
 - ② 15-17 = \otimes
 - ③ 10-17 = \otimes
 - ④ 17-17 = \otimes
- \Rightarrow End \approx mid-1

start mid End
15 15 16

- ① 19-15 = 4
 - ② 15-15 = \otimes
 - ③ 10-15 = \otimes
 - ④ 17-15 = 2
- $\Rightarrow 6$ mid

<문제> 정렬된 배열에서 특정 수의 개수 구하기: 문제 설명

- n개의 원소를 포함하고 있는 수열이 오름차순으로 정렬되어 있다. 이때 이 수열에서 x가 등장하는 횟수를 계산해라.

EX)

- 수열 {1, 1, 2, 2, 2, 2, 3}이 있을 때, x=2라면, 현재 수열에서 값이 2인 원소가 4개이므로 4를 출력한다.
- 단, 이 문제는 시간 복잡도 $O(\log N)$ 으로 알고리즘을 설계하지 않으면 시간 초과 판정을 받는다.

<문제 조건>

[입력 조건]

- 첫째 줄에 N과 x가 정수 형태로 공백으로 구분되어 입력된다.
($1 \leq N \leq 1,000,000$), ($-10^9 \leq x \leq 10^9$)
- 둘째 줄에 N개의 원소가 정수 형태로 공백으로 구분되어 입력된다.
($-10^9 \leq$ 각 원소의 값 $\leq 10^9$)

[출력 조건]

- 수열의 원소 중에서 값이 x인 원소의 개수를 출력한다. 단, 값이 x인 원소가 하나도 없다면 -1을 출력한다.

[입력 예시1]

```
7 2
1 1 2 2 2 2 3
```

[출력 예시1]

```
4
```

-> python(정렬된 배열에서 특정 수의 개수 구하기.py)