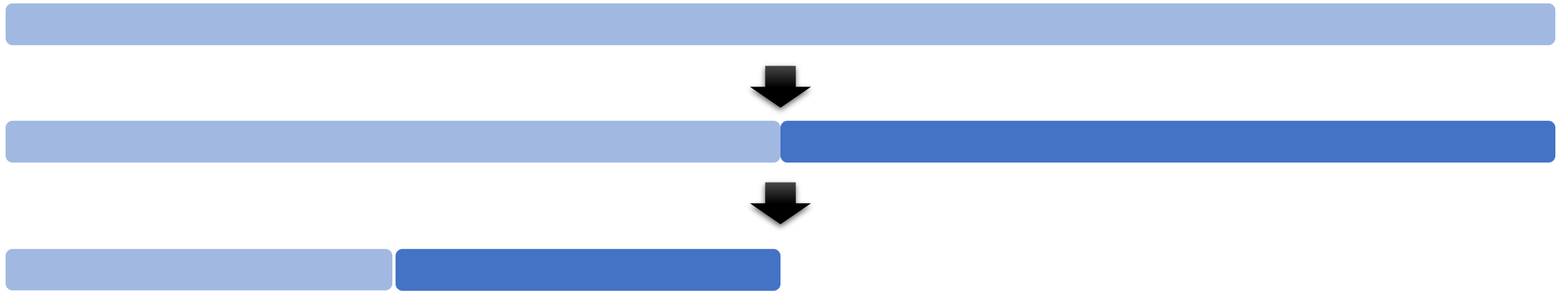


알면 쉬운
이진 탐색(binary search)

이진 탐색의 정의

Binary = 2진법의, 두 부분으로 이뤄진



→ 주어진 자료를 계속 반으로 나누어 가며 검색하는 방법

이진 탐색의 특징

- 데이터 정렬 필수
- 반복문으로 구현 추천
- 데이터 크기와 무관한 효율적 탐색
- 마지막에 찾는 경우라도 $O(\log n)$

이진탐색 트리

이진 탐색을 응용하여 만든 자료구조

해시 테이블

해시 함수(hash function)를 사용하여 데이터를

데이터 크기가 크고, 정렬이 가능한 알고리즘 문제를 빠르게 풀어야 할 때 사용

파라메트릭 서치(parametric search)

- 특정한 조건을 만족하는 가장 알맞은 값을 빠르게 찾는 최적화 문제

Ex) 예산, 랜선 자르기...

풀이방법

이진탐색을 반복하여 구하고자 하는 값이 중간점보다 크거나 작다면, 시작점이나 끝점의 값을 중간점으로 갱신하여 범위를 줄여 나가는 방식으로 풀이

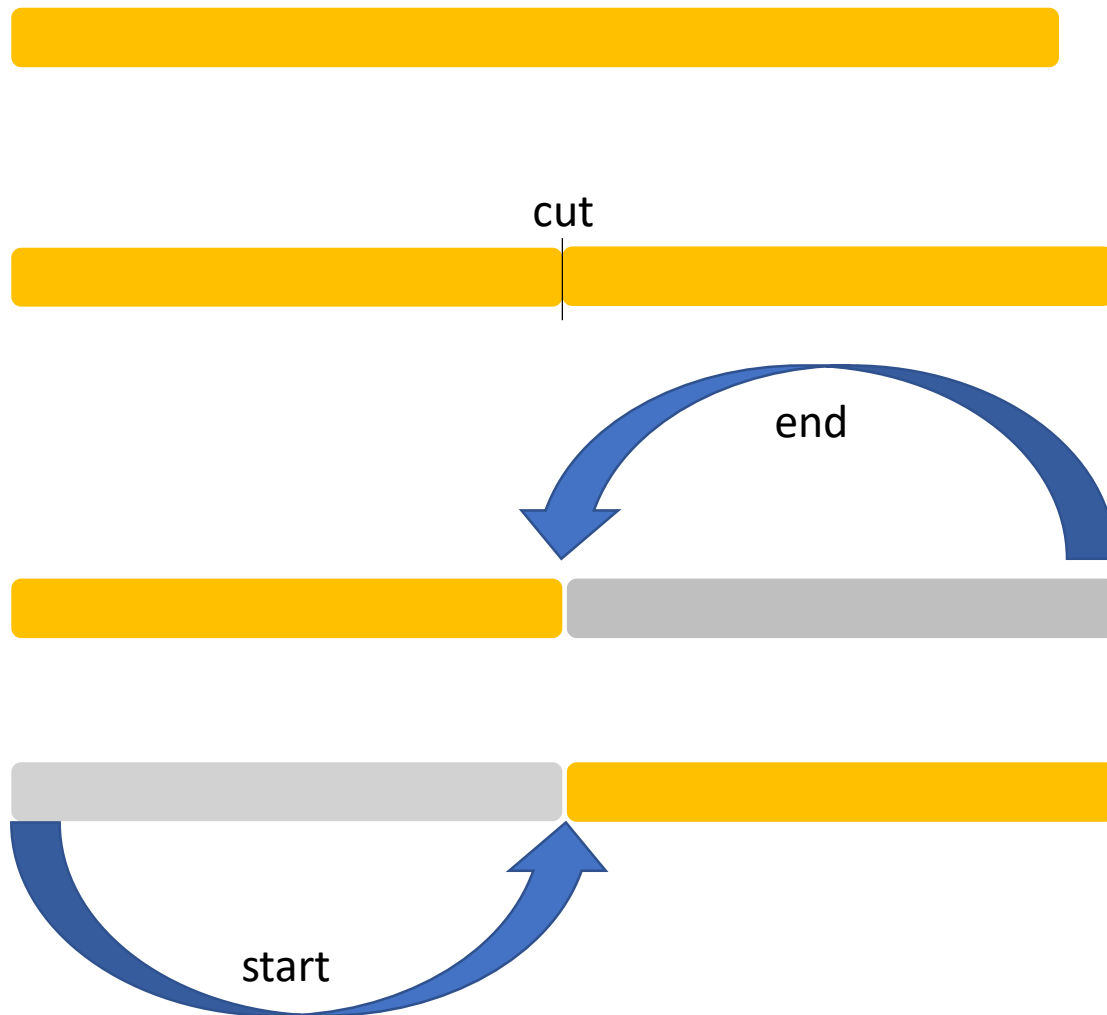
랜선 자르기

```
# 랜선의 개수와 자르고자 하는 길이 입력
n, cnt = map(int, input().split())
Lan = []
# 랜선에 대한 정보 입력
for i in range(n):
    Lan.append(int(input()))

# 이진탐색을 위한 시작점과 끝점 설정
start, end = 1, max(Lan)

# 이진탐색
while start <= end:
    cut = (start + end) // 2
    line = 0
    # 잘렸을때 랜선의 개수를 계산
    for i in Lan:
        line += i // cut
    # 랜선이 적은 경우 더 많이 잘려야하니(짧게 잘라야하니) 왼쪽 부분을 탐색
    if line < cnt:
        end = cut - 1
    # 랜선의 양이 너무 많은 경우 덜 잘려야하니 오른쪽부분 탐색
    else:
        start = cut + 1

# 정답 출력
print(end)
```



정렬된 배열에서 특정 수의 개수 구하기

문제

총 N 개의 정수가 주어졌을 때, 정수 v 가 몇 개인지 구하는 프로그램을 작성하시오.

입력

첫째 줄에 정수의 개수 N ($1 \leq N \leq 100$)이 주어진다. 둘째 줄에는 정수가 공백으로 구분되어져있다. 셋째 줄에는 찾으려고 하는 정수 v 가 주어진다. 입력으로 주어지는 정수와 v 는 -100보다 크거나 같으며, 100보다 작거나 같다.

출력

첫째 줄에 입력으로 주어진 N 개의 정수 중에 v 가 몇 개인지 출력한다.

예제 입력 1 복사

```
11
1 4 1 2 4 2 4 2 3 4 4
2
```

예제 출력 1 복사

```
3
```

<https://www.acmicpc.net/problem/10807>

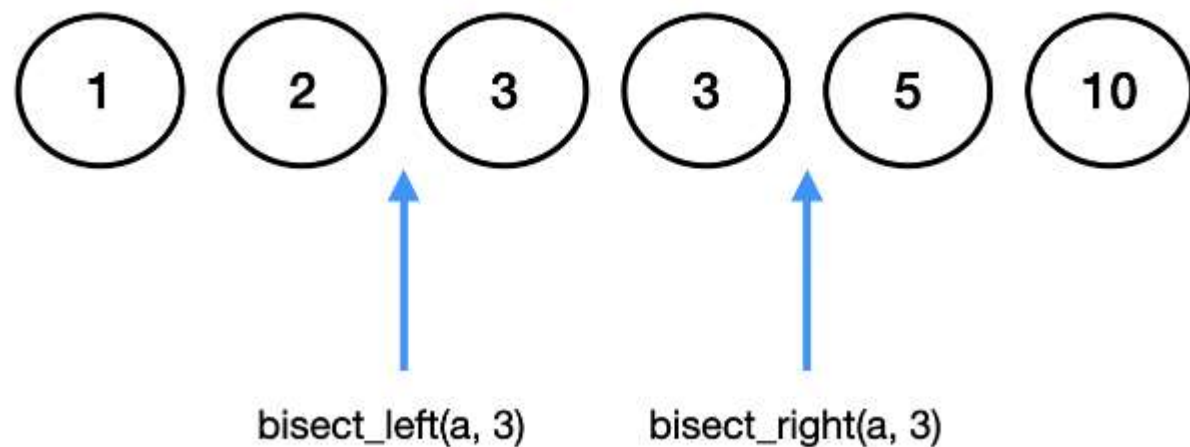
추천 라이브러리

- `Bisect_left(list, x)` :

정렬된 순서를 유지하면서 list에 x를 삽입할 가장 **왼쪽** 인덱스를 반환

- `Bisect_right(list, x)`

정렬된 순서를 유지하면서 list에 x를 삽입할 가장 **오른쪽** 인덱스를 반환



출력 값
2
4

예제 입력 1 복사

```
11
1 4 1 2 4 2 4 2 3 4 4
2
```

```
from bisect import bisect_left, bisect_right

def count(li, left, right):
    right_i = bisect_right(li, right)
    left_i = bisect_left(li, left)
    return right_i - left_i

n = int(input())
li = sorted(list(map(int, input().split())))
a = int(input())
print(count(li, a, a))
```

예제 출력 1 복사

```
3
```

정렬

[1, 1, 2, 2, 2, 3, 4, 4, 4, 4]

함수

Right_i = 2

Left_i = 5

Return = 3

Bisect 활용 문제

가장 긴 증가하는 부분 수열 3

<https://www.acmicpc.net/problem/12738>

문제

수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열을 구하는 프로그램을 작성하시오.

예를 들어, 수열 A = {10, 20, 10, 30, 20, 50} 인 경우에 가장 긴 증가하는 부분 수열은 A = {**10**, **20**, 10, **30**, 20, **50**} 이고, 길이는 4이다.

추천 반례

11

10 50 97 62 41 80 10 50 35 41 42

⇒ 4

5

5 4 3 2 1

⇒ 1

```

from bisect import bisect_left
import sys
input = sys.stdin.readline

n = int(input())
li = list(map(int, input().split()))
# 수열을 저장하는 리스트
li_2 = [-1000000001]

for i in li:
    # 직전에 저장한 수열보다 크면 추가
    if li_2[-1] < i:
        li_2.append(i)
    else:
        # 아니라면 해당 위치에 넣어준다
        # 작아져도 변경해주기때문에 꼭 필요
        which = bisect_left(li_2, i)
        li_2[which] = i

print(len(li_2)-1)

```

li_2 : 나올 수 있는 수 중 제일 작은 수 -1

For문에서 li 순서대로 li_2에 가장 최근에 들어온 수와 비교하면서 크다면 li_2에 넣어주기

작을 때 li_2에 들어온 수 중 자신의 오름차순에 맞게 넣어주어라

li_2에 하나를 넣고 시작했으니 -1

0	1	2	3
-1000000001	10	50	97

0	1	2	3
-1000000001	10	41	62

0	1	2	3	4
-1000000001	10	41	50	80

0	1	2	3	4
-1000000001	10	35	41	80

list

0	1	2	3	4
-1000000001	10	35	41	42