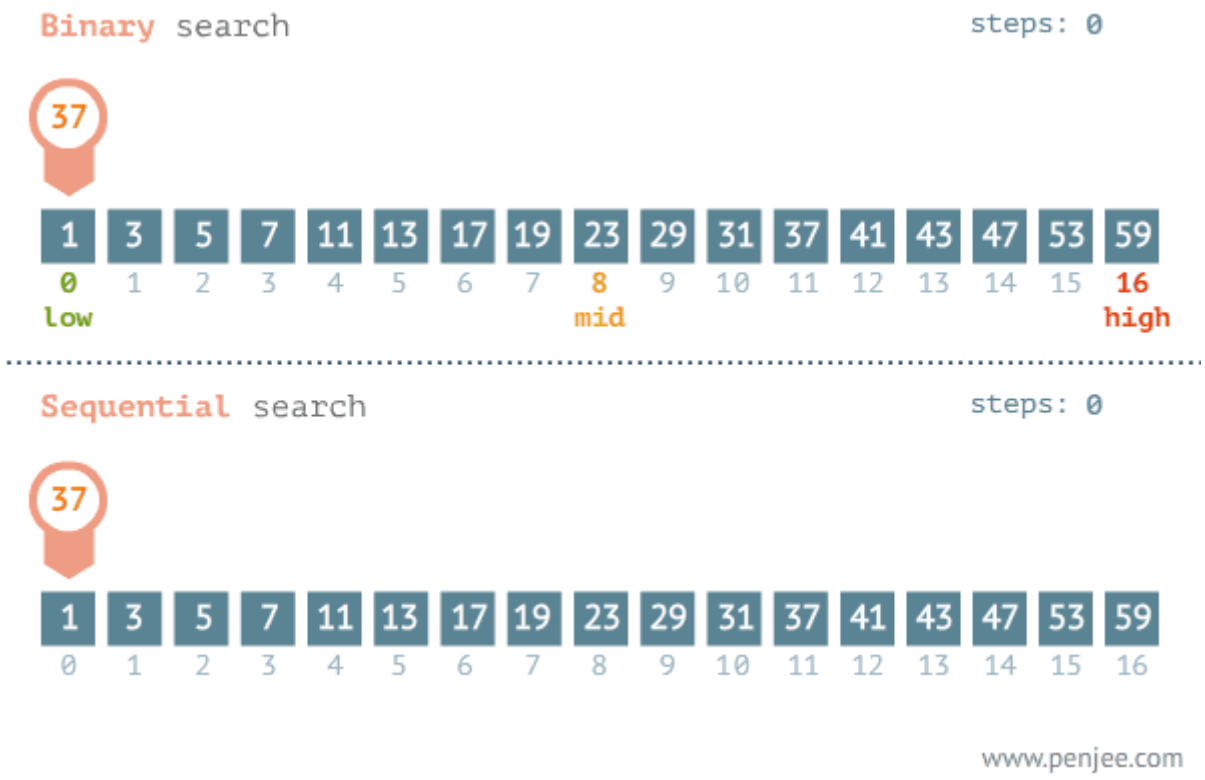
이진탐색

이진 탐색

- 정렬된 리스트에서 검색 범위를 반으로 줄여 나가면서 검색 값을 찾는 알고리즘
- 정렬된 리스트에서만 사용가능
- 대신 속도가 빠르다. 시간 복잡도 `O(logN)`

https://blog.penjee.com/wp-content/uploads/2015/04/binary-and-linear-search-animations.gif



재귀함수로 구현

```
# 재귀함수로 구현한 이진탐색
def binary_search(array, target, start, end):
    if start > end:
        return None
    mid = (start + end) // 2

    # 원하는 값 찾은 경우 인덱스 반환
    if array[mid] == target:
        return mid
    # 원하는 값이 중간점의 값보다 작은 경우 왼쪽 부분(절반의 왼쪽 부분) 확인
    elif array[mid] > target:
        return binary_search(array, target, start, mid - 1)
    # 원하는 값이 중간점의 값보다 큰 경우 오른쪽 부분(절반의 오른쪽 부분) 확인
    else:
        return binary_search(array, target, mid + 1, end)


n, target = list(map(int, input().split()))
array = list(map(int, input().split()))

result = binary_search(array, target, 0, n - 1)
if result is None:
    print('원소가 존재 X')
else:
    print(result + 1)
```

반복문으로 구현

```
# 반복문으로 구현한 이진 탐색
def binary_search(array, target, start, end):
    while start <= end:
        mid = (start + end) // 2

        # 원하는 값 찾은 경우 인덱스 반환
        if array[mid] == target:
            return mid
        # 원하는 값이 중간점의 값보다 작은 경우 왼쪽 부분(절반의 왼쪽 부분) 확인
        elif array[mid] > target:
            end = mid - 1
        # 원하는 값이 중간점의 값보다 큰 경우 오른쪽 부분(절반의 오른쪽 부분) 확인
        else:
            start = mid + 1

    return None


n, target = list(map(int, input().split()))
array = list(map(int, input().split()))
```

```
result = binary_search(array, target, 0, n - 1)
if result is None:
    print('원소가 존재 X')
else:
    print(result + 1)
```

Python 이진탐색 라이브러리

```
from bisect import bisect_left, bisect_right

a = [1,2,3,4,4,8]
x = 4
# 찾은 위치를 왼쪽부터 카운트 한 인덱스 반환
print(bisect_left(a,x))    # 3
# 찾은 위치를 오른쪽부터 카운트한 인덱스 반환
print(bisect_right(a,x))   # 4
```

https://velog.io/@woo0_hooo/python-Bisect-함수란

이분탐색으로 LIS 풀기

최대 증가 부분 수열 문제

백준 12738 가장 긴 증가하는 부분 수열 3

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

이진탐색으로 LIS풀기

A={10, 20, 10, 20, 20, 50}

가장 긴 증가하는 부분수열찾기.

접근방식: 주어진 리스트를 맨앞에서부터 순회하며

이진탐색을 거쳐, 새로운 리스트에 담기

이때, 들어오는 값이 증가하는 값이면 리스트맨뒤에추가

만약 증가하는값이 아니라면 이진탐색을통해 내 앞에서 위치에 넣는다.

A = [10, 20, 10, 20, 20, 50] LIS = [ ]

- step1 10

  LIS = [10]

- step2 20

  LIS = [10 20]

- step3 10

  bisect_left(LIS, 10) = 0.

  LIS = [10, 20]

- step4 30

  LIS = [10, 20, 30]

- step5 20

  bisect_left(LIS, 20) = 1

  LIS = [10, 20, 30]

- step6 50

  LIS = [10, 20, 30, 50]

len(LIS) = 4.

▼ 코드

```
# 가장 긴 증가하는 부분 수열 3 LIS
# 이진 탐색으로 풀기
import sys
input = sys.stdin.readline
```

```
from bisect import bisect_left #이진탐색 코드, 같은 수일 경우 왼쪽 index를 돌려준다

N = int(input())
A = list(map(int, input().split()))
lis = list()

for i in A:
    k = bisect_left(lis, i) #자신이 들어갈 위치 k
    # print(k)
    if len(lis) <= k: #i가 가장 큰 숫자라면
        lis.append(i)
    else:
        lis[k] = i #자신보다 큰 수 중 최솟값과 대체
print(len(lis))
```

문제

공통

- BOJ _2470 두 용액

  https://www.acmicpc.net/problem/2470

문제집 링크

https://www.acmicpc.net/problemset?sort=ac_desc&algo=12