

# 투 포인터

강의자: 정지호

# 투 포인터가 필요한 이유?

1부터 5까지의 수가 담겨있는 배열에서 합이 5인 부분 연속 수열의 개수 구하기

1	2	3	4	5
---	---	---	---	---

# 투 포인터가 필요한 이유?

투 포인터를 모르는 경우 -> 이중 for문을 사용해 하나하나 구한다.

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

# 투 포인터가 필요한 이유?

이중 for문 사용 코드

```
numbers = [1, 2, 3, 4, 5]
count = 1 # 부분수열의 합이 5가 되는 경우의 수를 저장하는 변수

for i in range(5):
    sum = numbers[i]
    for j in range(i+1, 5):
        sum += numbers[j]
        if sum == 5:
            count += 1

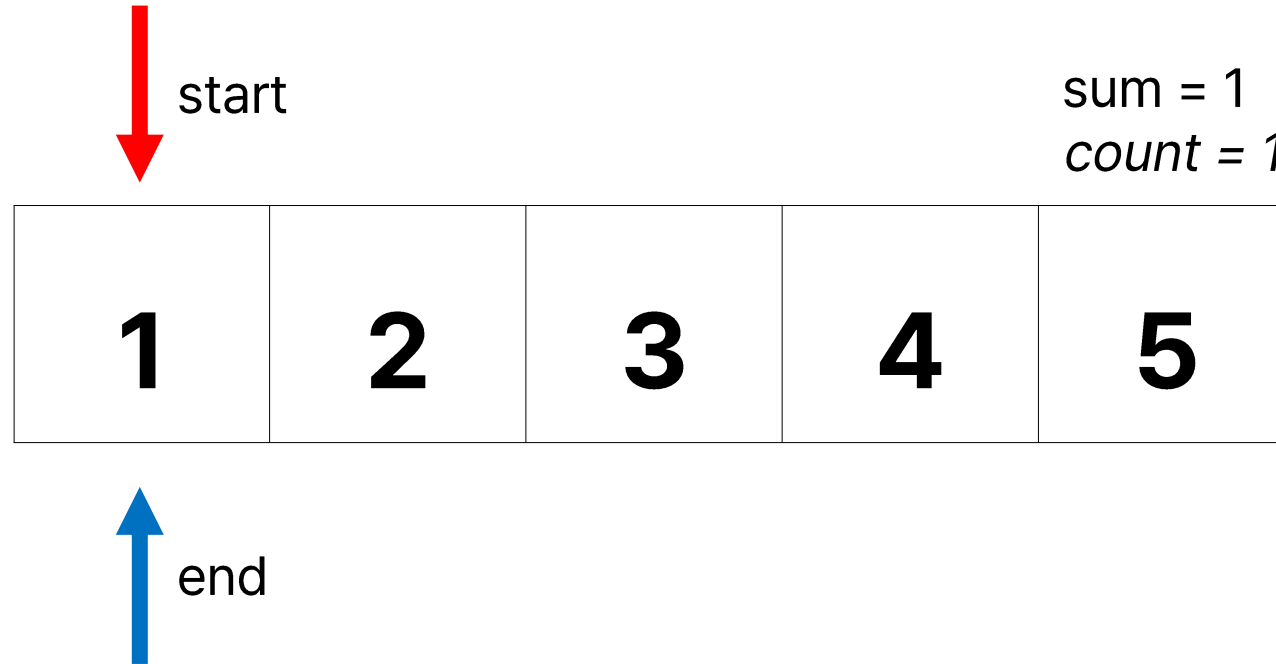
print(count)
```

시간복잡도  $O(n^2)$

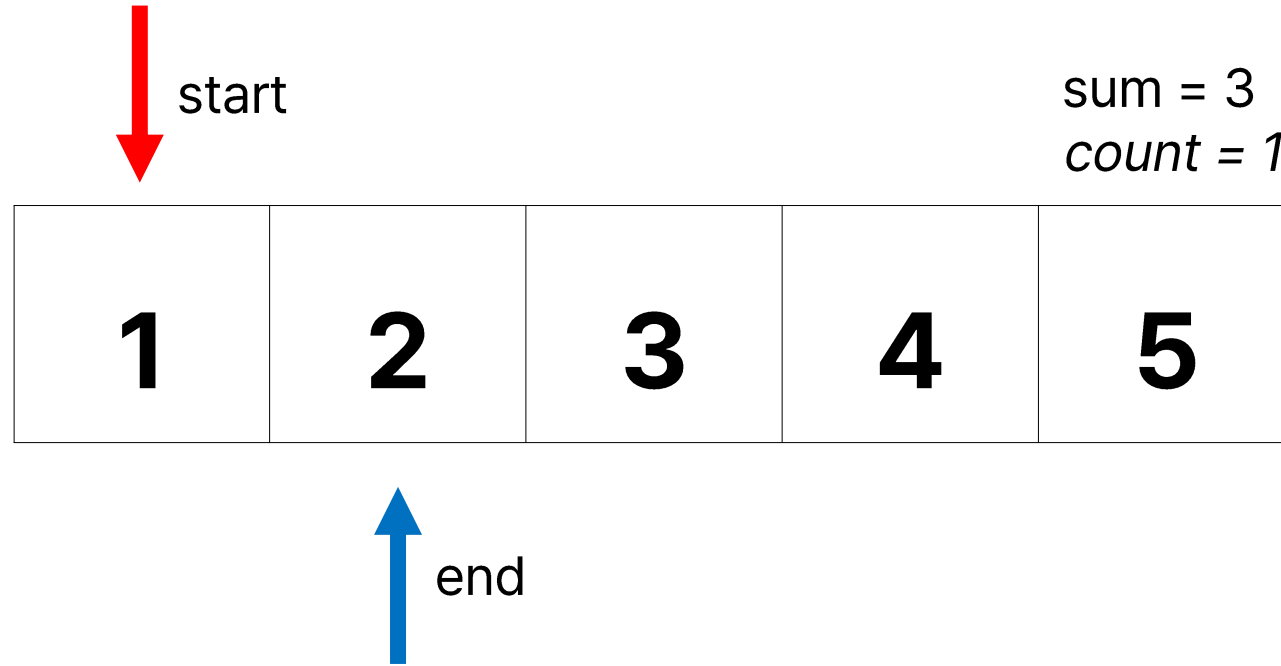
count = 1로 시작하는 이유?

배열에 5가 이미 있으므로 자기 자신 혼자만 있는 수열을 미리 저장함

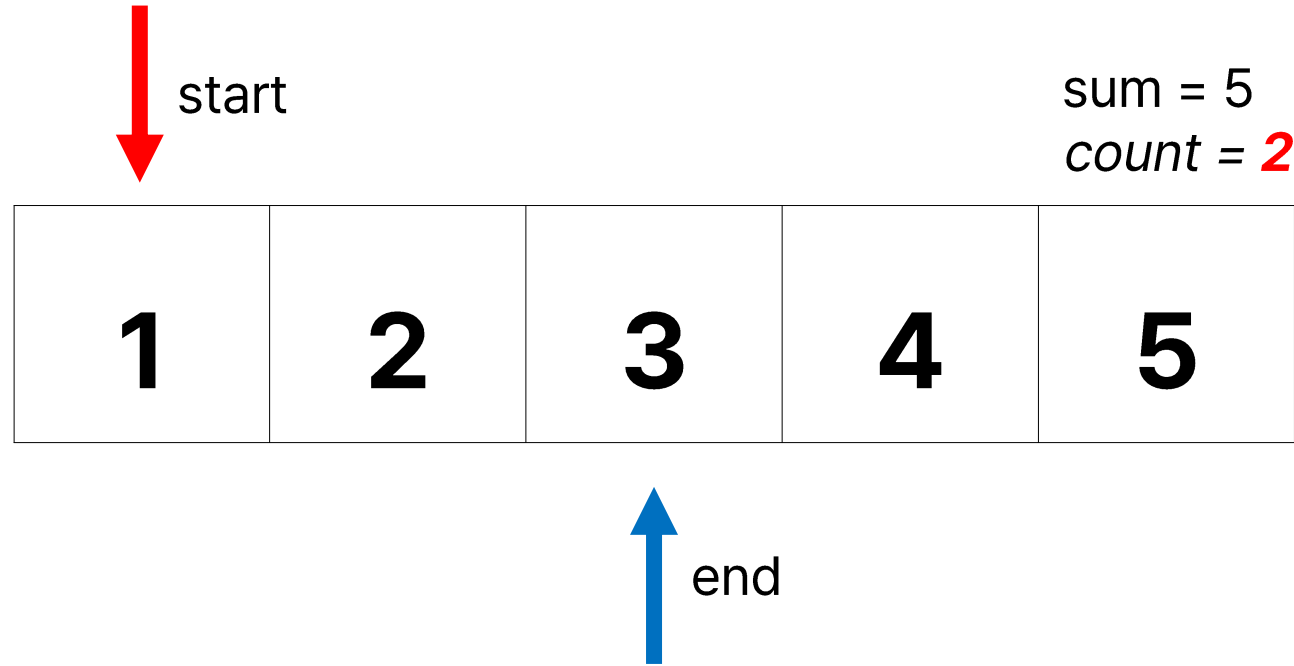
# 투 포인터 이해하기



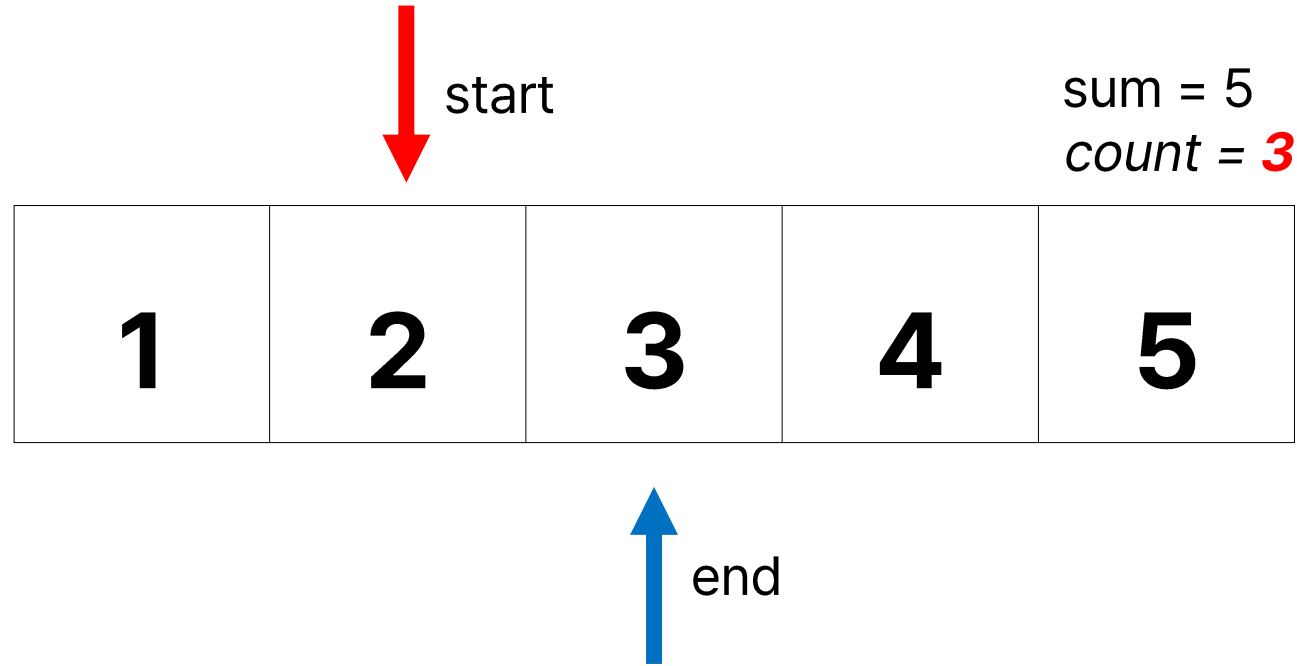
# 투 포인터 이해하기



# 투 포인터 이해하기

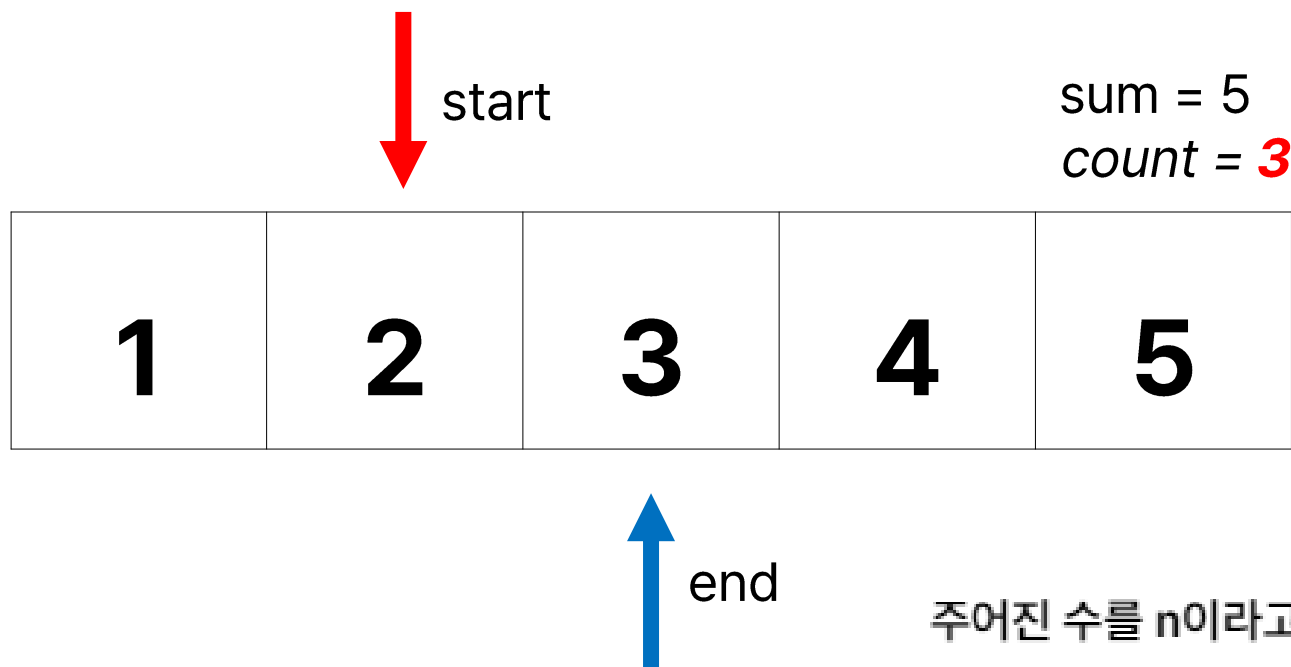


# 투 포인터 이해하기





# 투 포인터 이해하기



주어진 수를  $n$ 이라고 했을 때,

- $sum < n$ : `end_index` 증가, `sum` 값 변경
- $sum > n$ : `sum` 값 변경, `start_index` 증가
- $sum == n$ : `count` 증가, `end_index` 증가, `sum` 값 변경
- 마지막으로 `count` 출력

## 투 포인터 구현 코드

```
numbers = [1, 2, 3, 4, 5]
count = 1    # 부분수열의 합이 5가 되는 경우의 수를 저장하는 변수
sum = 1
start_index = 0
end_index = 0

while end_index < 4:
    if sum < 5:
        end_index += 1
        sum += numbers[end_index]
    elif sum > 5:
        sum -= numbers[start_index]
        start_index += 1
    else:
        count += 1
        end_index += 1
        sum += numbers[end_index]

print(count)
```

# 백준 1940번 주몽

## ✓ 문제

주몽은 철기군을 양성하기 위한 프로젝트에 나섰다.

그래서 야철대장을 통해 철기군이 입을 갑옷을 만들게 하였다.

야철대장은 주몽의 명에 따르기 위하여 연구에 착수하던 중 아래와 같은 사실을 발견하게 되었다.

갑옷을 만드는 재료들은 각각 고유한 번호를 가지고 있다.

갑옷은 두 개의 재료로 만드는데 두 재료의 고유한 번호를 합쳐서  $M(1 \leq M \leq 10,000,000)$ 이 되면 갑옷이 만들어 지게 된다.

야철대장은 자신이 만들고 있는 재료를 가지고 갑옷을 몇 개나 만들 수 있는지 궁금해졌다.

이러한 궁금증을 풀어 주기 위하여  $N(1 \leq N \leq 15,000)$ 개의 재료와  $M$ 이 주어졌을 때 몇 개의 갑옷을 만들 수 있는지를 구하는 프로그램을 작성하시오.

## ✓ 입력

첫째 줄에는 재료의 개수  $N(1 \leq N \leq 15,000)$ 이 주어진다. 그리고 두 번째 줄에는 갑옷을 만드는데 필요한 수  $M(1 \leq M \leq 10,000,000)$  주어진다. 그리고 마지막으로 셋째 줄에는  $N$ 개의 재료들이 가진 고유한 번호들이 공백을 사이에 두고 주어진다. 고유한 번호는 100,000보다 작거나 같은 자연수이다.

## ✓ 출력

첫째 줄에 갑옷을 만들 수 있는 개수를 출력한다.

## ✓ 입출력 예제

입력 1

```
6
9
2 7 4 1 5 3
```

출력 1

```
2
```

## 시간 제한 : 2초

- N의 최대 범위 : 15,000개

Q.  $O(N^2)$  사용 가능?

$$(15,000)^2 = 225,000,000$$

파이썬은 1초에 최대 1억 번의 연산을 할 수 있음.

Q.  $O(N \log N)$  사용 가능?

<b>2</b>	<b>7</b>	<b>4</b>	<b>1</b>	<b>5</b>	<b>3</b>
----------	----------	----------	----------	----------	----------

**M = 9**

2	7	4	1	5	3
---	---	---	---	---	---

**M = 9**

2	7	4	1	5	3
---	---	---	---	---	---

**M = 9**

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>7</b>
----------	----------	----------	----------	----------	----------

**M = 9**



<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>7</b>
----------	----------	----------	----------	----------	----------

**M = 9**



1	2	3	4	5	7
---	---	---	---	---	---

**M = 9**





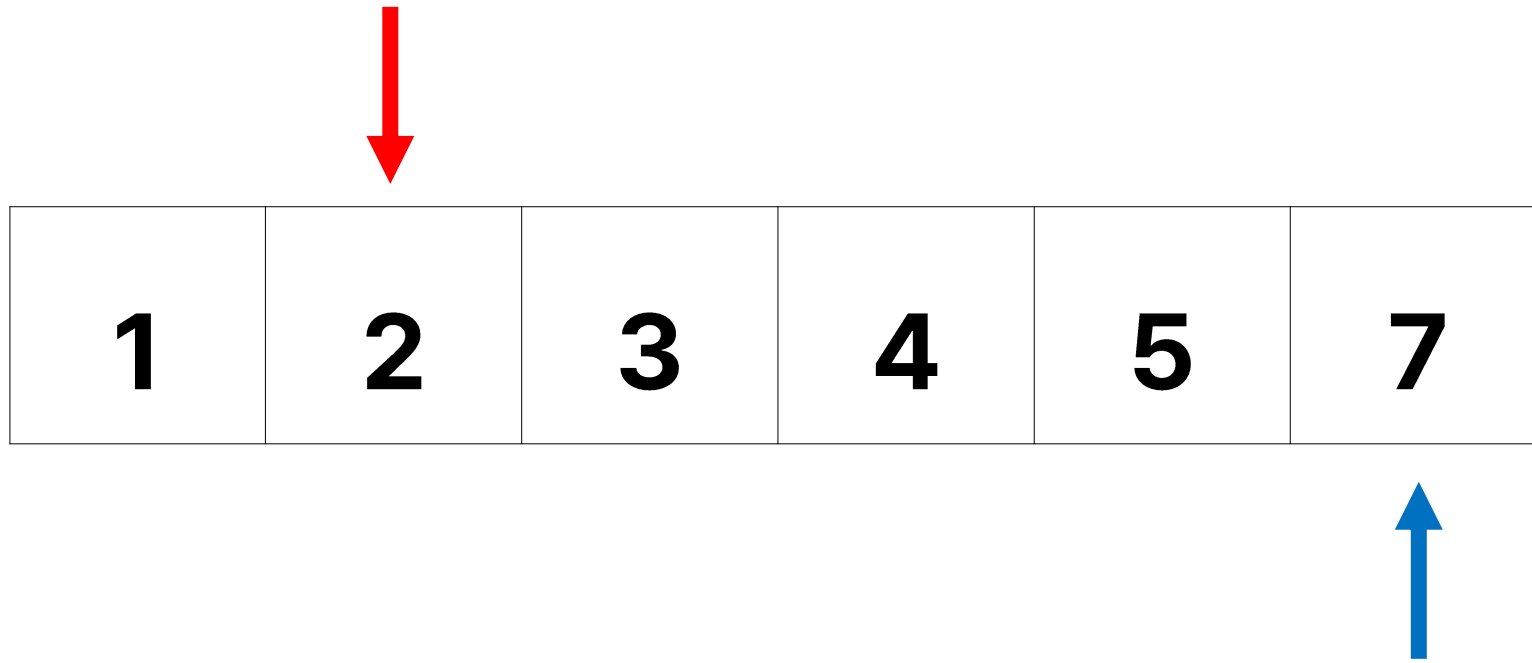
1	2	3	4	5	7
---	---	---	---	---	---



**M > sum**

**M = 9**

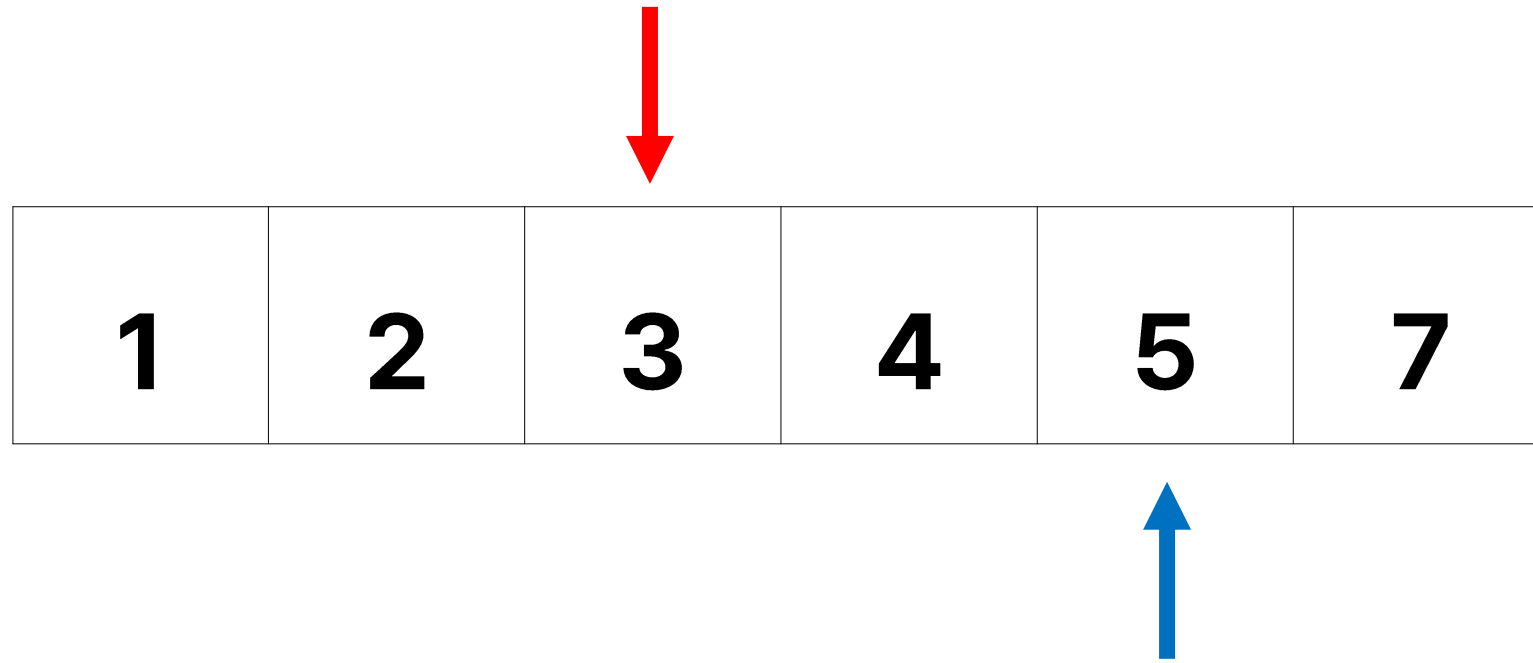
**sum = 8**



**M = sum**

**M = 9**

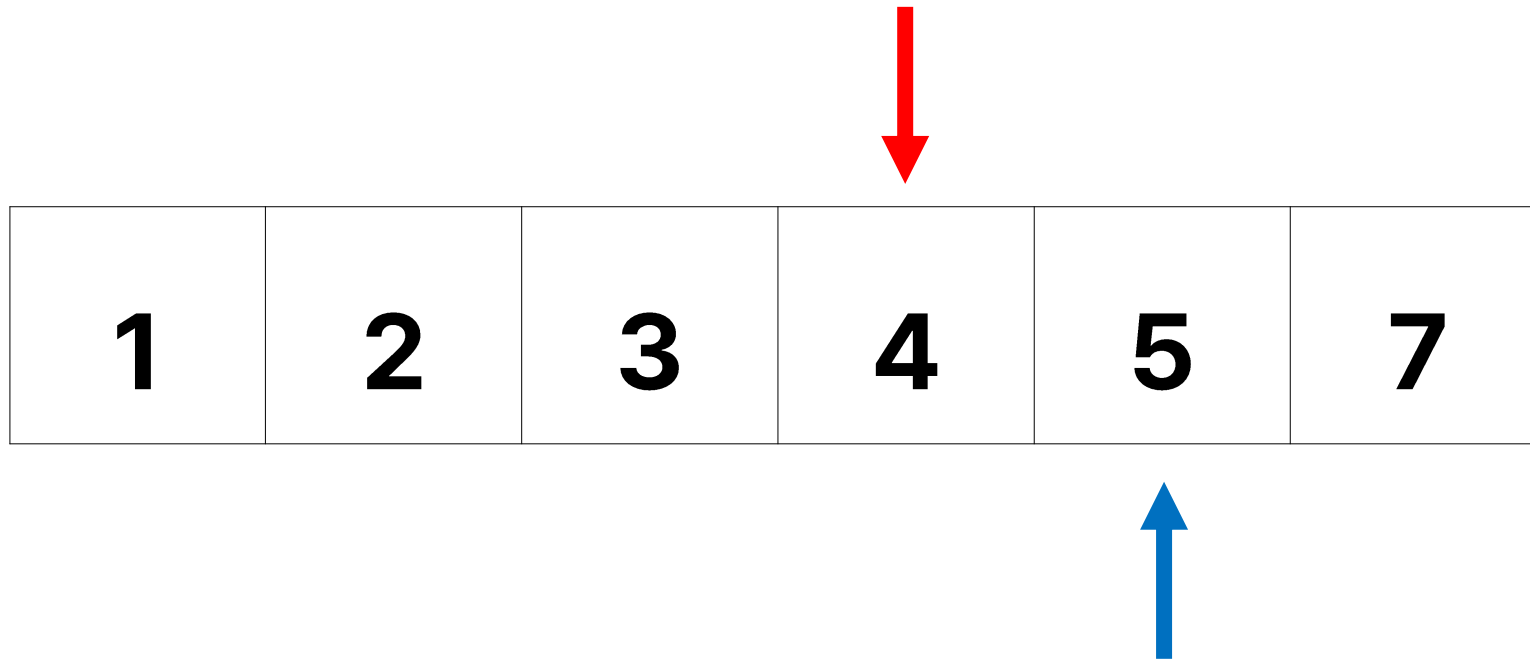
**sum = 9**



**M > sum**

**M = 9**

**sum = 8**



**M = sum**

**M = 9**

**sum = 9**

**(과제)**

- 백준 2018
- 백준 1940
- 백준 1253

**더 공부해보고 싶으면 풀어볼 문제**

- 2003
- 1644
- 1806
- 2230
- 1484
- 2038
- 2531
- 2096
- 2293