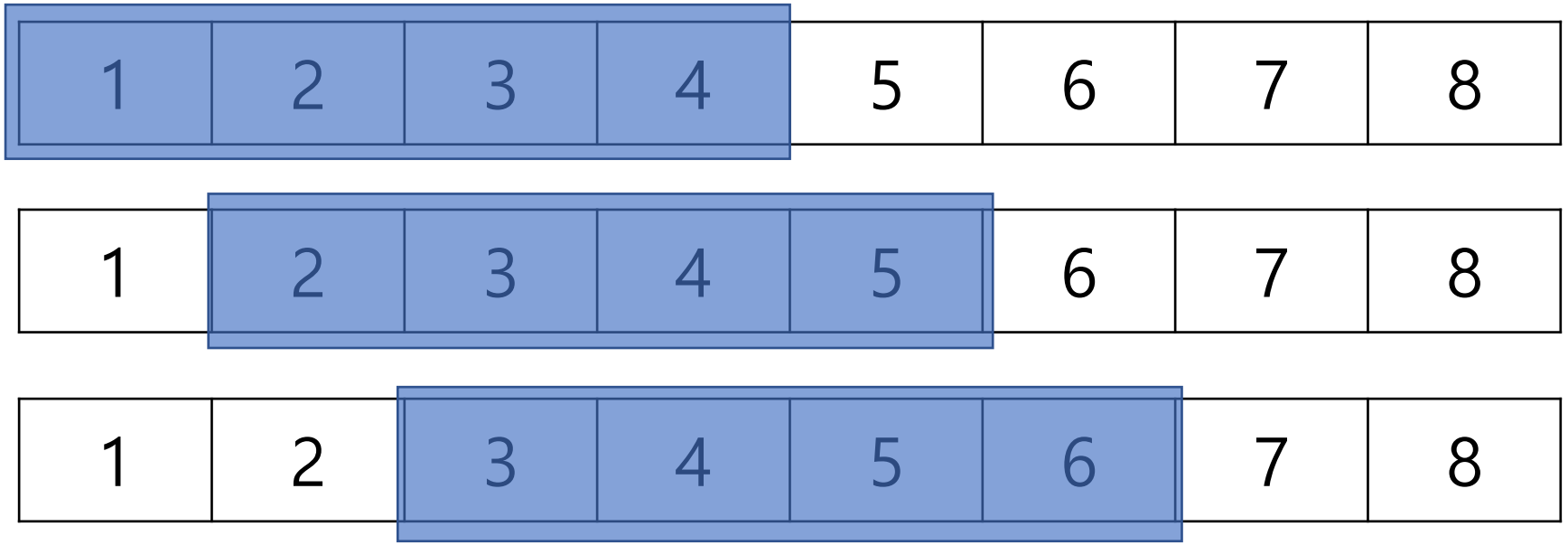
Algorithm

- 슬라이딩 윈도우

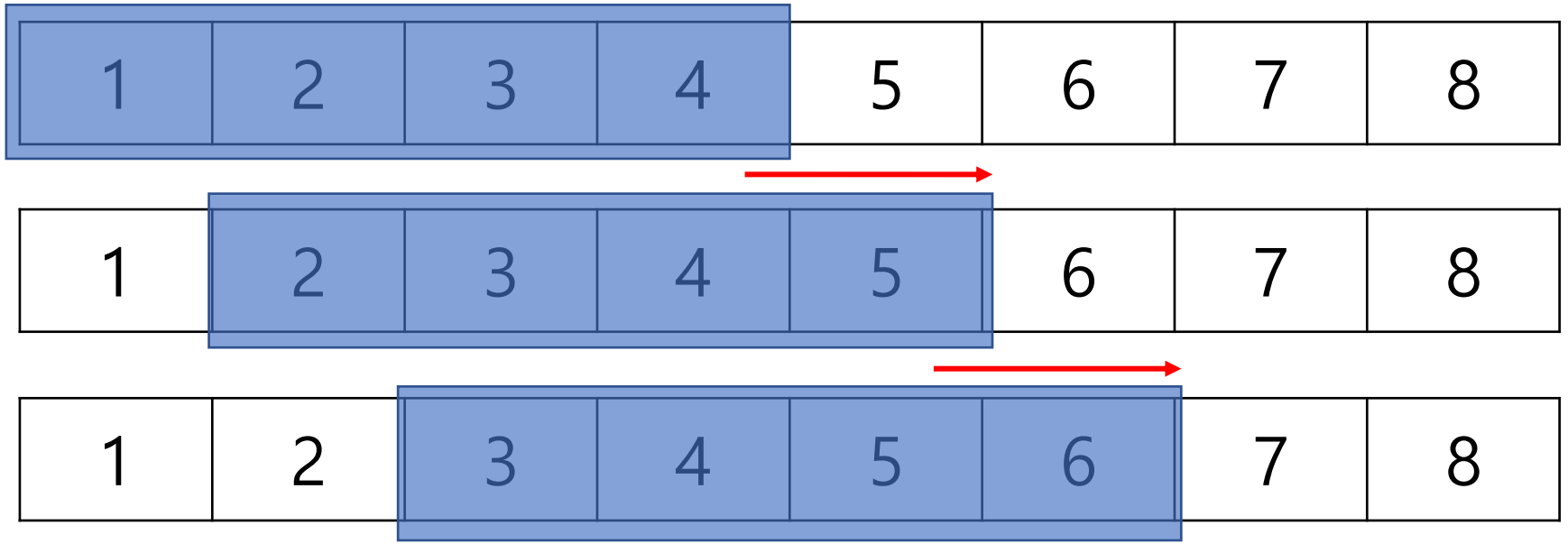
슬라이딩 윈도우

- 슬라이딩 윈도우?
  - 고정 사이즈의 윈도우가 이동하면서 윈도우 내에 있는 데이터를 이용해 문제를 푸는 것(투 포인터와 비슷한 느낌 but, 사이즈 고정)
  - 예시. 연속된 범위의 숫자 4개의 합이 가장 커지는 범위는?



슬라이딩 윈도우

- 예시. 연속된 범위의 숫자 4개의 합이 가장 커지는 범위는?



- 연속된 범위의 숫자 합을 구하기 위해 범위를 바꾸고, 다시 속한 숫자를 합하면?

슬라이딩 윈도우

- 예시. 연속된 범위의 숫자 4개의 합이 가장 커지는 범위는?

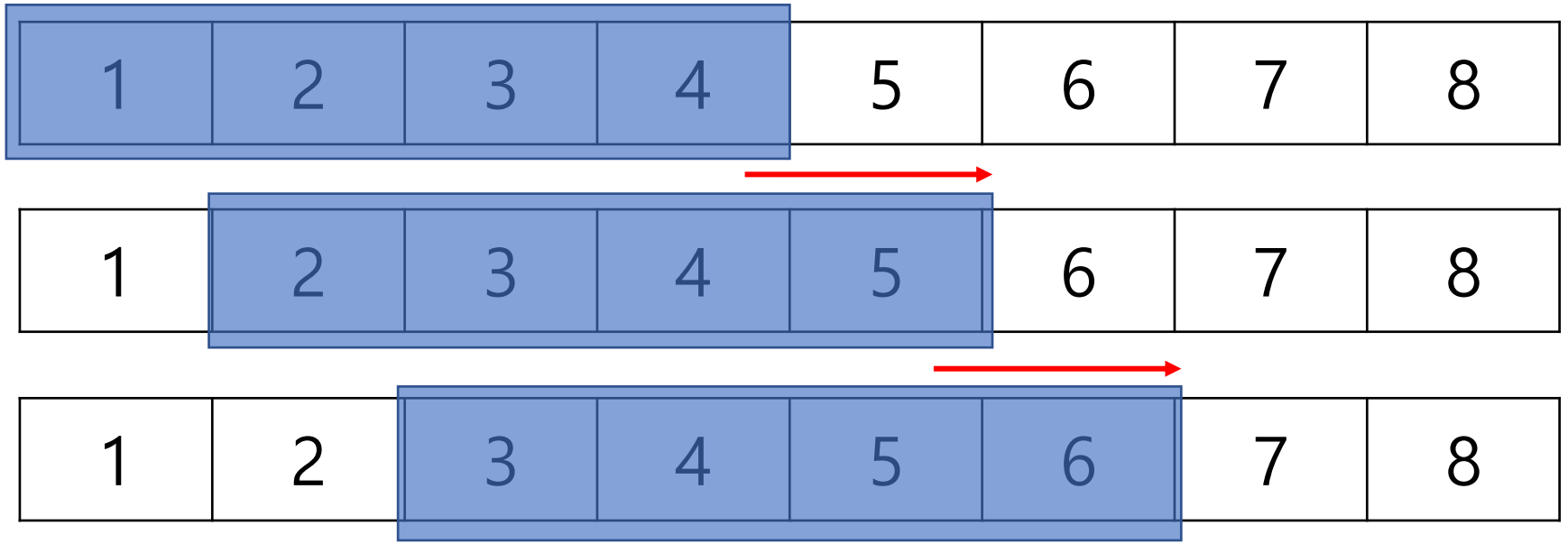
- 연속된 범위의 숫자 합을 구하기 위해 범위를 바꾸고, 다시 속한 숫자를 합하면?
  - 답을 구할 수 있지만 주어진 숫자 N의 크기가 커지면 시간 OVER!

```
static int test1() {
    int sum =0;
    for(int i=0; i<=N-M; i++) {
        int temp=0;
        for(int j=i; j<i+M; j++) {
            temp+=arr[j];
        }
        if(temp>sum) sum=temp;
    }
    return sum;
}
```

시간 복잡도 O(N*M)

슬라이딩 윈도우

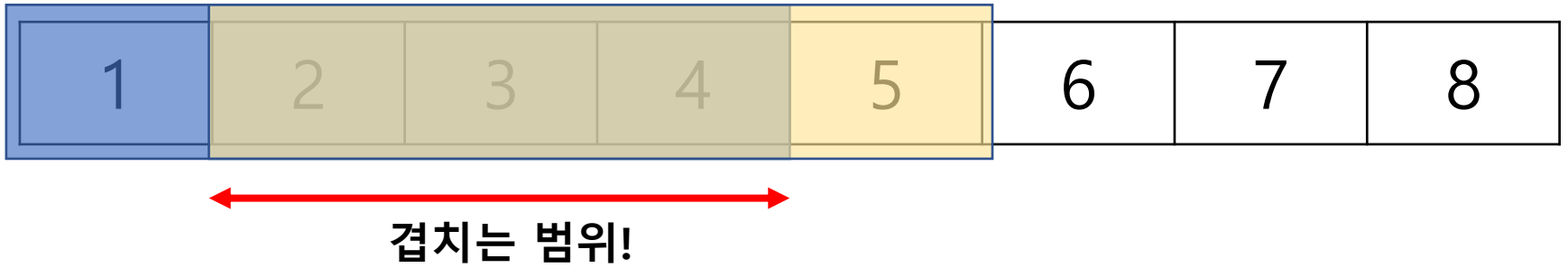
- 예시. 연속된 범위의 숫자 4개의 합이 가장 커지는 범위는?



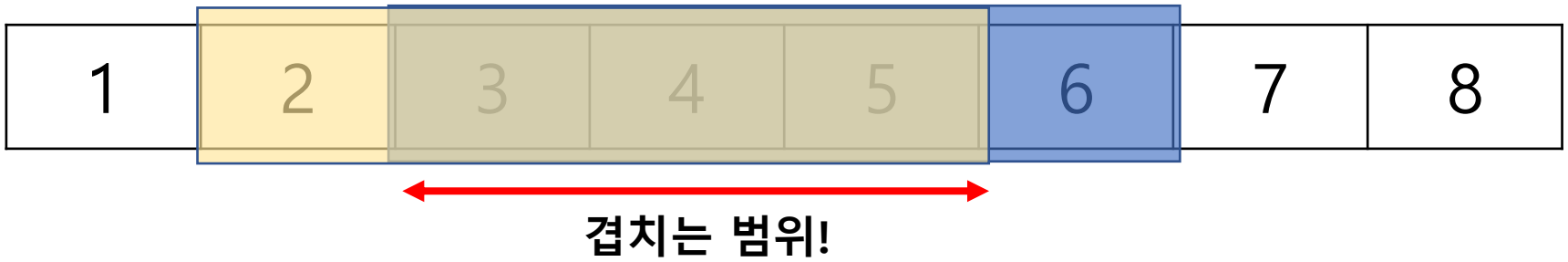
- **고정된 범위가 이동하는 부분에 집중!!**

슬라이딩 윈도우

- 1번째 범위 합을 구한 뒤 2번째 범위 합을 구하는 경우

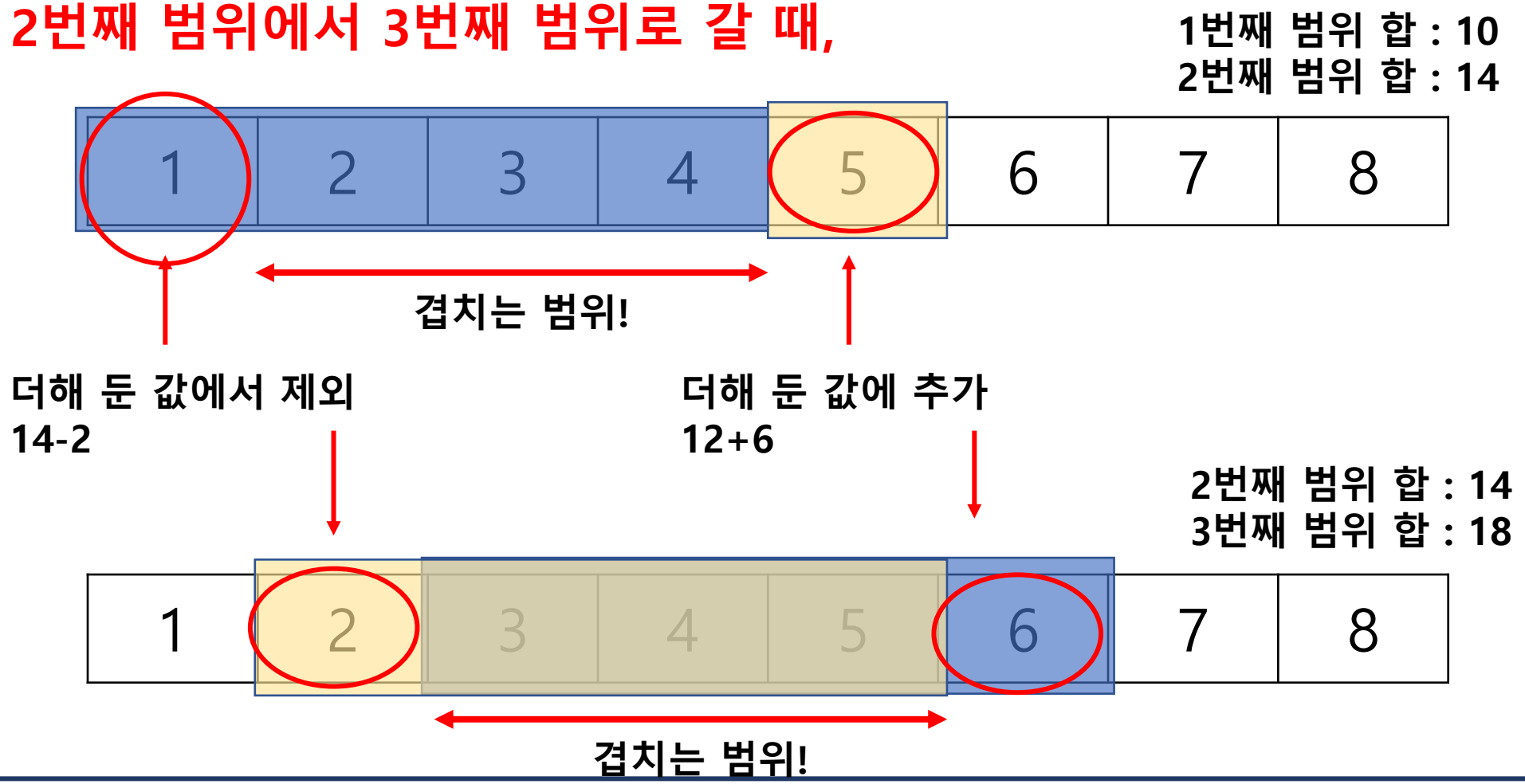


- 2번째 범위 합을 구한 뒤 3번째 범위 합을 구하는 경우



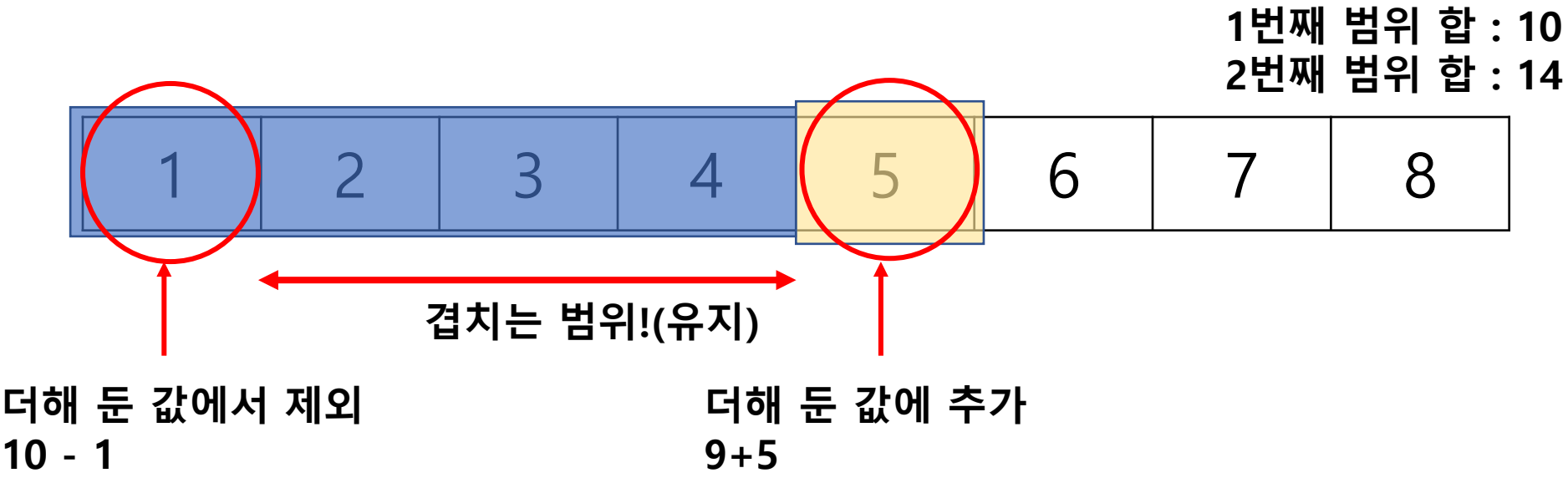
슬라이딩 윈도우

- 겹치는 범위를 제외한 실제 바뀌는 값들만 활용하기.
  - 2번째 범위에서 3번째 범위로 갈 때,



슬라이딩 윈도우

- 이런 형태로 고정된 범위(윈도우)에서 바뀌는 값들만 더하고 빼는 형태로 연산



슬라이딩 윈도우

- 이런 형태로 고정된 범위(윈도우)에서 바뀌는 값들만 더하고 빼는 형태로 연산

```
static int test2() {
    int sum=0;
    int temp=0;

    for(int i=0; i<M; i++) {
        temp+=arr[i];
        sum=temp;
    }
    for(int i=M; i<N; i++) {
        int j=i-M; //더해진 값 중 가장 먼저 더해진 값을 빼기 위한 변수
        temp = temp-arr[j]+arr[i];
        if(temp>sum) sum=temp;

    }
    return sum;
}
```

시간 복잡도 O(N)

초기 값 셋팅

바뀌는 값들만 작업

슬라이딩 윈도우

- **고정된 범위**가 주어지고,
  해당 범위의 합, 조건 충족 여부 확인 등의 문제를 풀 때 사용 고려
- anagram 처럼 범위 안에 주어진 문자열/숫자 등이 모두 포함 되어있는지 확인 할 때 사용 고려 등
- 투 포인터랑 혼동하지 않도록 주의!!