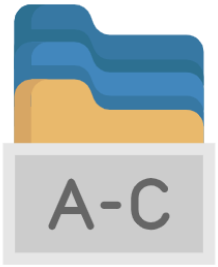
투 포인터(two pointer)



'투 포인터'는 정렬된 배열에서 두 개의 포인터(인덱스)를 이용하여 해당 값들과 원하는 값을 비교한 뒤 포인터를 조작하여 원하는 결과를 얻어내는 기법이다. 대표적인 예로 '배열 내 합이 $O(N^2)$가 되는 순서쌍 찾기'가 있다. 투 포인터 기법으로 해결하기 전에 가장 간단한 방법인 이중 반복문 사용으로 배열 내 합이 $O(N)$가 되는 순서쌍을 찾아보자!

```
for(int i = 0; i<n; ++i)
    for(int j = i +1; j<n; ++j)
        if(seq[i] + seq[j] == S)
            cout << seq[i] << '+' << seq[j] << '=' << S << '₩n';
```

간단하게 구현할 수 있지만, 이중 반복문을 이용하는 경우 $O(N^2)$의 시간이 걸리므로 데이터가 크면 클수록 시간 초과가 발생할 가능성이 매우 높아집니다. 하지만 투 포인터를 이용하면 정렬되지 않은 배열의 경우 $O(NlogN)$이미 정렬된 배열의 경우 $O(N)$시간에 이러한 순서쌍들을 찾을 수 있다. 다음과 같이 정렬된 배열을 통해 예를 들어 보면,

1	3	5	6	9	11	12	16	17	19	22	25	28

두 개의 포인터를 이용하여 양 끝 지점을 가리킵니다. 왼쪽 끝의 포인터를 L 오른쪽 끝의 포인터를 R 이라고 했을때,

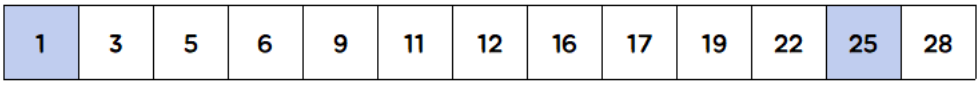
1	3	5	6	9	11	12	16	17	19	22	25	28

이제 L<R 을 만족하는 동안 다음과 같은 작업을 반복한다.

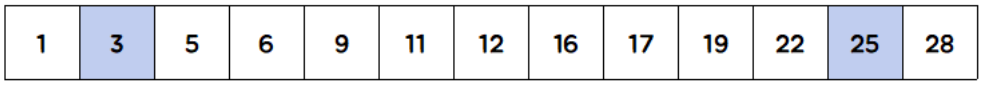
- 두 개의 포인터가 가리키고 있는 원소의 합이 S 보다 큰 경우 R 을 감소한다.
- 두 개의 포인터가 가리키고 있는 원소의 합이 S 보다 작은 경우 L 을 증가한다.
- 두 개의 포인터가 가리키고 있는 원소의 합이 S 와 같다면 L 을 증가시키고 R 을 감소시키거나 작업을 종료한다.

먼저 첫 번째 경우 두 원소의 합이 S 보다 크기 때문에 오름차순으로 정렬된 배열에서 R 을 감소시킨다면 그 값이 작아지게 된다. 그 다음 두 번째 경우, 두 원소의 합이 S 보다 작기 때문에 오름차순으로 정렬된 배열에서 L 을 증가시킨다면 그 값이 커지게 된다. 마지막 세 번째 경우에서 어떠한 답을 출력하건 상관없다면 작업을 종료해도 무방하지만, 특정한 조건에 맞는 답을 구하거나 (예를 들면 중복된 원소가 있는지 등) 모든 경우의 수를 구해야 한다면 경우에 맞는 적절한 작업을 해줘야만 한다.

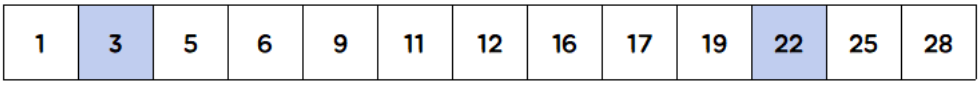
위의 예제에서 두 원소의 합이 27 인 경우를 찾아볼까요? 먼저 두 개의 포인터가 가리키고 있는 원소의 합은 29 이다. 이 값은 찾고자 하는 27 보다 큰 값이므로 R 을 하나 감소시켜 준다.



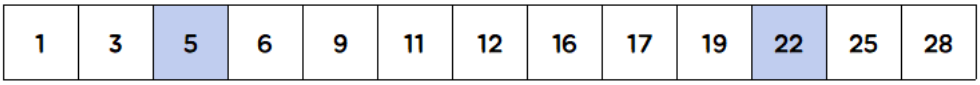
이후 두 원소의 합이 27 보다 작은 값이므로 L 을 하나 증가시킨다.



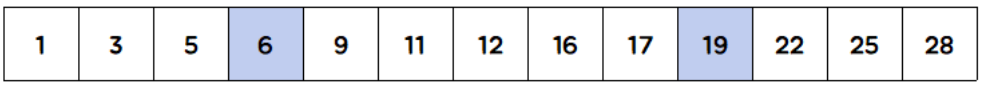
두 원소의 합이 28 이므로 다시 R 을 하나 감소시킨다.



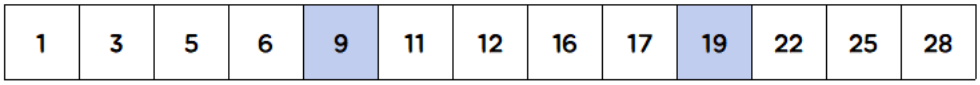
두 원소의 합이 25 이므로 L 을 하나 증가시킨다.



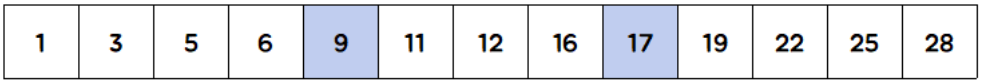
이제 우리가 찾고자 했던 합이 27 이 되는 두 개의 원소를 찾았지만, 예제의 경우 배열에 중복된 원소가 없으니 조건에 맞는 또 다른 순서쌍이 존재할 수 있으니 진행을 멈추지 않습니다. L 을 하나 증가, R 을 하나 감소시킨다.



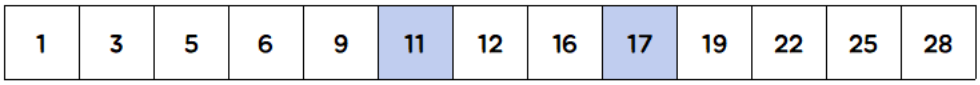
두 원소의 합이 25 이므로 L 을 하나 증가시킨다.



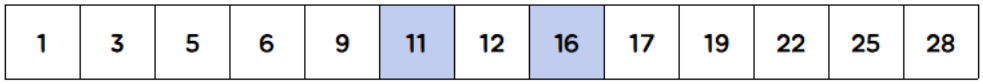
두 원소의 합이 28 이므로 R 을 하나 감소시킨다.



두 원소의 합이 26 이므로 L 을 하나 증가시킨다.



두 원소의 합이 28 이므로 R 을 하나 감소시킨다.



찾고자 했던 합이 27 이 되는 두 개의 원소 쌍을 하나 더 찾았습니다. 다시 다른 순서쌍을 찾기 위해 L 을 하나 증가, R 을 하나 감소시킨다.

이제 두 포인터가 하나의 원소를 가리키고 있고 이때 원소의 합은 24 이므로 찾고자 했던 순서쌍이 아니다. 이제 더 이상 $L \leq R$을 만족하지 않으니 작업을 종료한다. 이 작업을 수행하는 코드는 아래의 예시를 통해 볼 수 있다. N 개의 중복 없는 원소를 입력받아 합이 S 가 되는 순서쌍의 개수를 찾는 코드이다. 위 예제와 함께 찬찬히 살펴보자!