

알고리즘 특강 **투**포인터

배열의 모든 경우를 순회할 필요 없이, 포인터 두개로 O(N)만에 탐색하는 방법입니다. <u>최근에 등장하는 빈도</u>가 늘어나고 있는 유형입니다.

시작부터 문제?



Silver 3 - 구간 합 구하기 4 (#11659)

요약

• 수 N개가 주어졌을 때, i번째 수 부터 j번째 수 까지의 합을 구하는 프로그램을 작성하시오.

제약조건

- 수의 개수의 범위는 1 <= N <= 100,000 이다.
- 합을 구하는 횟수의 범위는 1 <= M <= 100,000 이다.

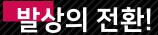




웰이 다 더하던...

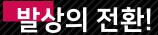
2 5 8 11 12 24 1 3

1번부터 8번까지의 합을 20만 번만 출력해줘!





● Sum(N, M): N번 인덱스 부터 M번 인덱스 까지의 합.





- Sum(N, M): N번 인덱스 부터 M번 인덱스 까지의 합.
- Sum(1, M): 1번 인덱스 부터 M번 인덱스 까지의 합.





- Sum(N, M): N번 인덱스 부터 M번 인덱스 까지의 합.
- Sum(1, M): 1번 인덱스 부터 M번 인덱스 까지의 합.

```
Sum(N, M) = dt[N] + dt[N + 1] ··· + dt[M]
= (dt[1] + dt[2] ··· + dt[N - 1]) + dt[N] + dt[N + 1] ··· + dt[M] - (dt[1] + dt[2] ··· + dt[N - 1])
= Sum(1, M) - Sum(1, N - 1)
```





- Sum(N, M): N번 인덱스 부터 M번 인덱스 까지의 합.
- Sum(1, M): 1번 인덱스 부터 M번 인덱스 까지의 합.
- Sum(N, M) = dt[N] + dt[N + 1] ··· + dt[M]
 = (dt[1] + dt[2] ··· + dt[N 1]) + dt[N] + dt[N + 1] ··· + dt[M] (dt[1] + dt[2] ··· + dt[N 1])
 = Sum(1, M) Sum(1, N 1)

Sum 배열을 만들어 꾸던 될것 같네!







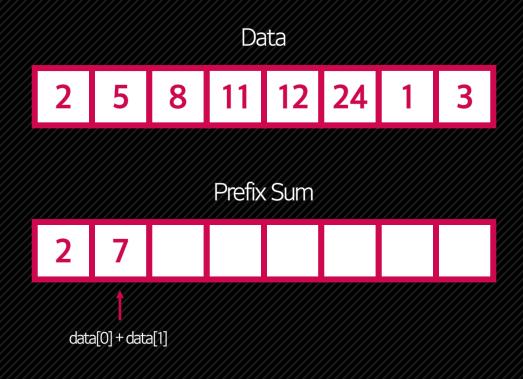


Prefix Sum



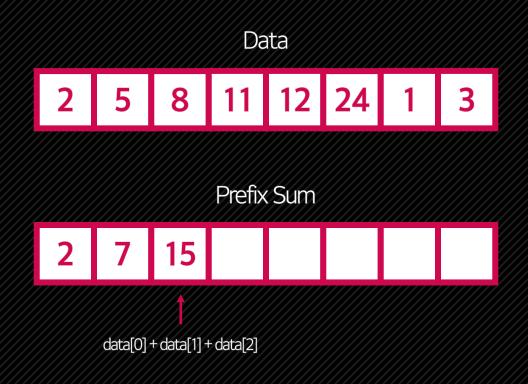






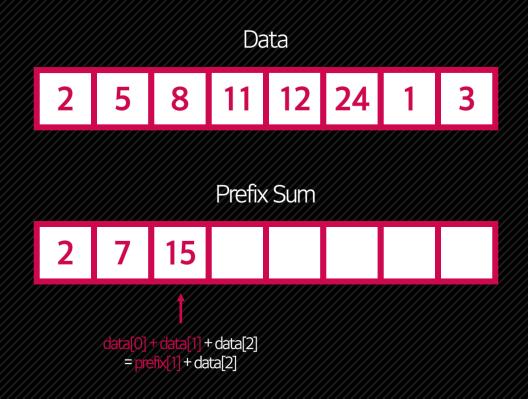
















```
Data
```



Prefix Sum

```
2 7 15 26 38 62 63 66
```

```
set prefix[0] <- data[0]
for idx = 1 ... length(data)
   set prefix[idx] <- prefix[idx - 1] + data[idx]</pre>
```







Two Pointers

- 1차원 배열에서 두개의 포인터를 조작하여 원하는 것을 얻는 기법.
- O(N²)가 걸릴 수 있는 배열 선택 문제를 O(N)로 해결할 수 있음.



문제로 연습해보자.



Gold 5 - 두 용액 (#2470)

요약

- 두 용액을 섞어 산성도를 최대한 0에 가깝게 만들려고 한다.
- 두 개의 서로 다른 용액을 혼합하여 0에 가깝게 만드는 두 용액을 찾는 프로그램을 작성하시오.

제약조건

- 용액의 수 N의 범위는 1 <= N <= 100,000 이다.
- 용액의 산성도의 범위는 -1,000,000,000 <= N_i <= 1,000,000,000 이다.







Gold 5 - 두용액 (#2470)

요약

- 두 용액을 섞어 산성도를 최대한 0에 가깝게 만들려고 한다.
- 두 개의 서로 다른 용액을 혼합하여 0에 가깝게 만드는 두 용액을 찾는 프로그램을 작성하시오.

제약조건

- 용액의 수 N의 범위는 1<= N <= 100,000 이다.
- 용액의 산성도의 범위는 -1,000,000,000 <= N; <= 1,000,000,000 이다.

접근

● 이제 완탐이 왜 안되는지는 알겠죠?







Gold 5 - 두용액 (#2470)

요약

- 두 용액을 섞어 산성도를 최대한 0에 가깝게 만들려고 한다.
- 두 개의 서로 다른 용액을 혼합하여 0에 가깝게 만드는 두 용액을 찾는 프로그램을 작성하시오.

제약조건

- 용액의 수 N의 범위는 1<= N<= 100,000 이다.
- 용액의 산성도의 범위는 1,000,000,000 <= N; <= 1,000,000,000 이다.

접근

- 이제 완탐이 왜 안되는지는 알겠죠?
- 뭔가 제일 작은 것과 제일 큰 것을 비교하고 싶지 않나요?







Gold 5 - 두용액 (#2470)

요약

- 두 용액을 섞어 산성도를 최대한 0에 가깝게 만들려고 한다.
- 두 개의 서로 다른 용액을 혼합하여 0에 가깝게 만드는 두 용액을 찾는 프로그램을 작성하시오.

제약조건

- 용액의 수 N의 범위는 1<= N<= 100,000 이다.
- 용액의 산성도의 범위는 -1,000,000,000 <= N <= 1,000,000,000 이다.

접근

- 이제 완탐이 왜 안되는지는 알겠죠?
- 뭔가 제일 작은 것과 제일 큰 것을 비교하고 싶지 않나요?
- 0보다 크면? → 줄이지!
- → 0보다 작으면? → 늘리자!

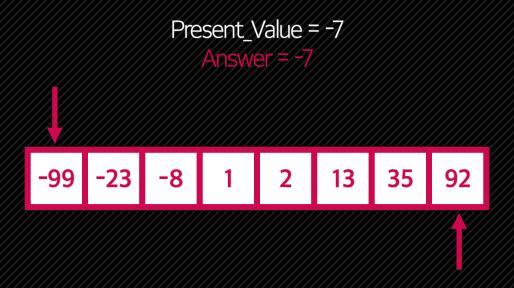


Gold 5 - 두 용액 (#2470)



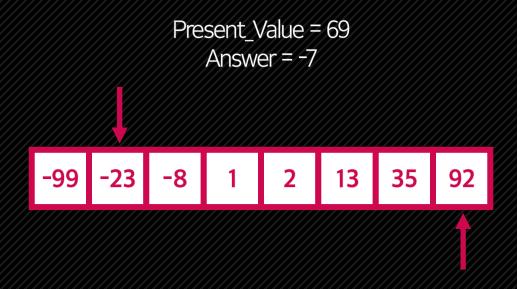


Gold 5 - 두용액 (#2470)



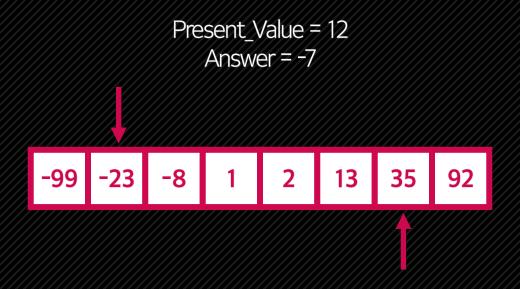


/◇ Gold 5 - 두용액 (#2470)



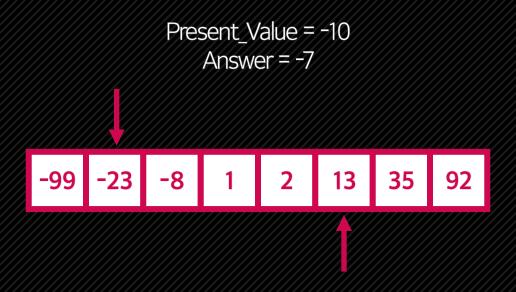


/◇ Gold 5 - 두용액 (#2470)



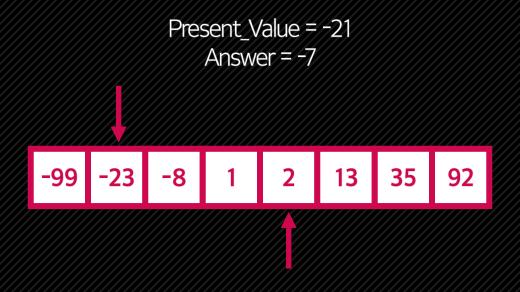






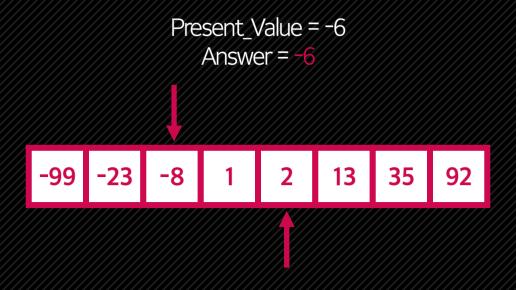


/◇ Gold 5 - 두용액 (#2470)



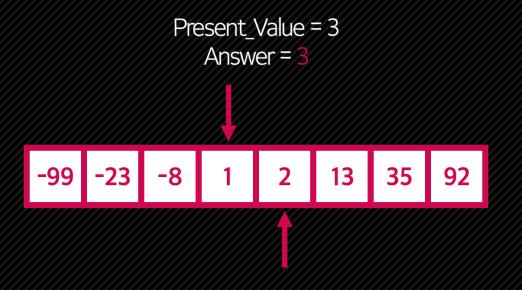








/◇ Gold 5 - 두용액 (#2470)



부분합 안녕!





Gold 4 - 부분합 (#1806)

요약

- 10,000 이하의 자연수로 이루어진 길이 N짜리 수열이 주어진다.
- 연속된 부분합 중 그 합이 S 이상이 된 것 중에서 가장 짧은 수열의 길이를 출력하는 프로그램을 작성하시오.

제약조건

- N의 범위는 10 <= N <= 100,000 이다.
- S의 범위는 0 <= S <= 100,000,000 이다.





```
Data
```



Prefix Sum

```
2 7 15 26 38 62 63 66
```

```
set prefix[0] <- data[0]
for idx = 1 ... length(data)
   set prefix[idx] <- prefix[idx - 1] + data[idx]</pre>
```







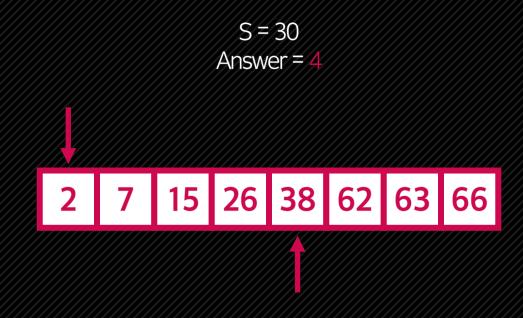




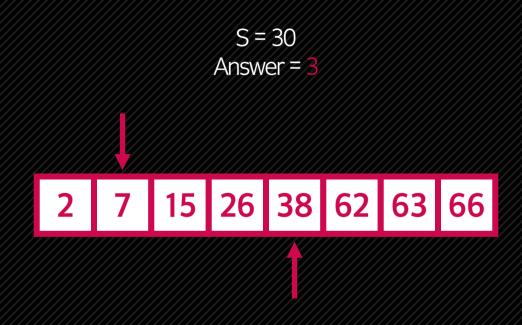




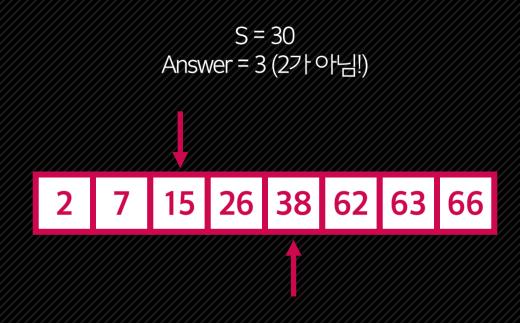




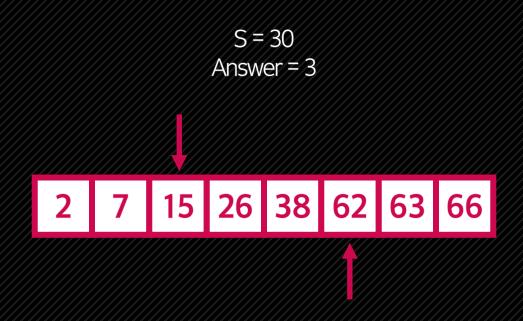




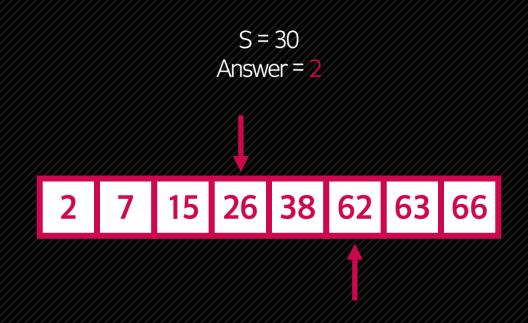




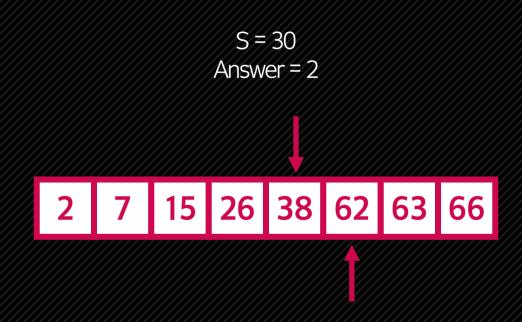






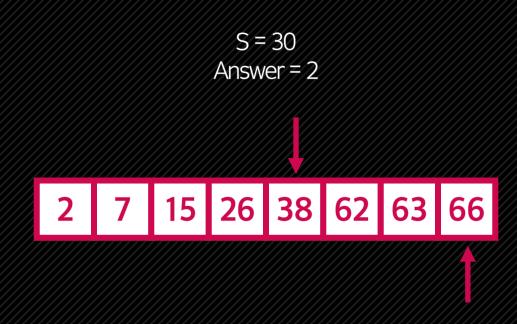








꼭 시작점이 달라야 할까?



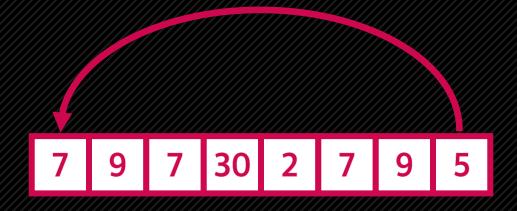






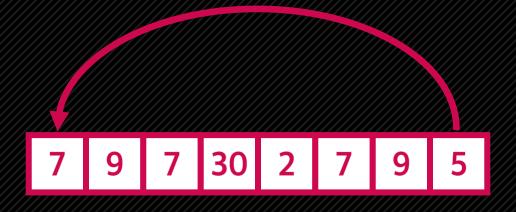


Silver 1 - 회전 초밥 (#2531)



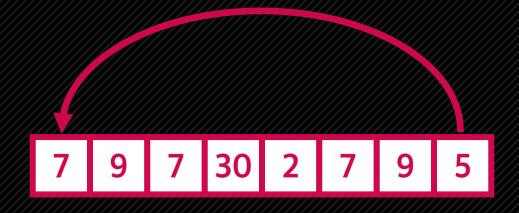
● 만약 연속된 K개의 범위에 쿠폰에 해당하는 값이 없다면?





- 만약 연속된 K개의 범위에 쿠폰에 해당하는 값이 없다면?





- 만약 연속된 K개의 범위에 쿠폰에 해당하는 값이 없다면?
- → 무조건 범위와 다른 종류이니 결과 값+1
- 만약 연속된 K개의 범위에 쿠폰에 해당하는 값이 있다면?
- 무조건 범위와 겹치게 되므로 그대로 출력



Silver 1 - 회전 초밥 (#2531)

함께 풀어봅시다.



● 원형이라 평면으로 계산하면 잘리는 부분은 어떡하지??



Silver 1 - 회전 초밥 (#2531)

7 9 7 30 2 7 9 5 7 9 30

- 원형이라 평면으로 계산하면 잘리는 부분은 어떡하지??
- →→→ 뒤에 더 붙여주면 해결!







- 두 포인터의 인덱스 차이는 동일하게 유지해야 함
- Sliding Window라고 부르는 알고리즘!









📂 Level 3 - 단속 카메라

요약

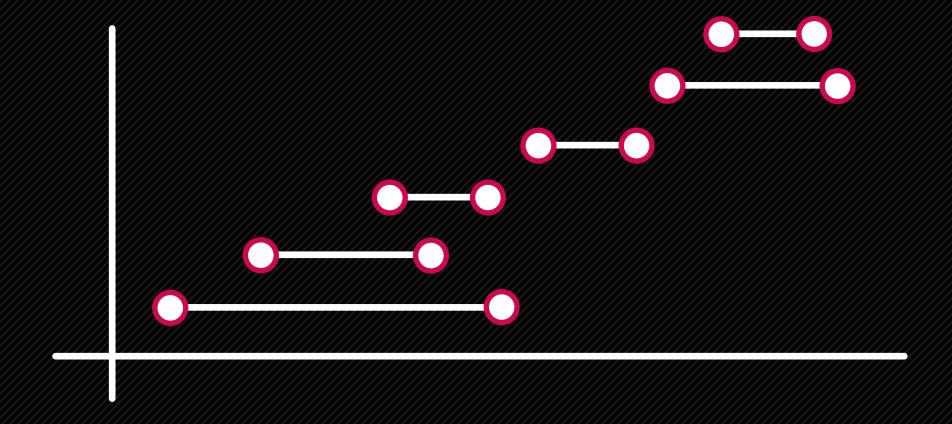
- 고속도로를 이동하는 모든 차량이 최소 1번 이상 단속 카메라를 만나도록 설치하려고 한다.
- 각 차의 진입점과 진출점이 주어질 때, 1번 이상 카메라를 만나도록 하는 최소의 카메라 수를 구하는 프로그램을 작성하시오.

제약조건

- 차량의 수는 1 <= N <= 10,000 이다.
- 진입점, 진출점의 범위는 -30,000 <= 1, 0 < 30,000 이다.

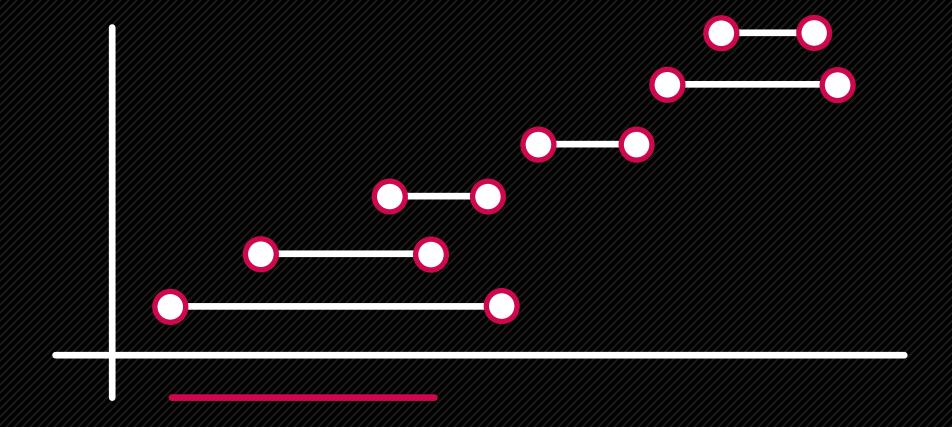






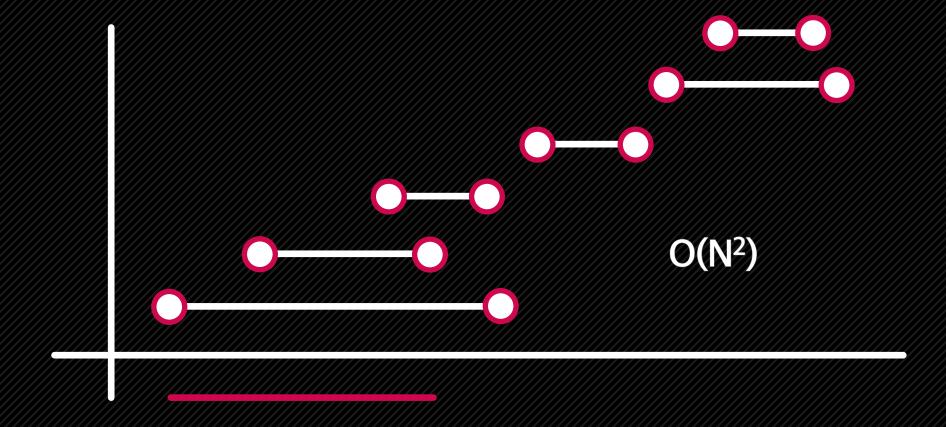
















- Level 3 보석 쇼핑 (2020 카카오 인턴십 코딩테스트)
- ★ Level 3 광고 삽입 (2021 kakao blind recruitment 1차 코딩테스트)

추가 추천 문제

- ✓ Silver 1 귀여운 라이언
 - 정보를 어떻게 바꿔야 할까요?
- - Three Pointer는 없습니다. 수의 범위를 보면서 생각을 해보시고 투 포인터로 바꿔보세요.
- - 소수를 판별하기 위한 에라토스테네스의 체 알고리즘을 공부하셔야 합니다. 필요한 주제니까 이참에 공부해보죠.

</s>;

"Any question?"