

Two Pointers Algorithm

⇒ Two Pointers 알고리즘은 리스트(배열)에 순차적으로 접근할 때, 2개의 점의 위치를 기록하면서 처리하는 알고리즘을 의미한다.

대표적인 문제로는 특정한 합을 가지는 부분 연속 수열 찾기를 볼 수 있다.

① 특정한 합을 가지는 부분 연속 수열 찾기

⇒ 리스트가 주어졌을 때, 부분 연속 수열 중에서 특정한 합을 가지는 수열의 개수를 출력하는 문제이다.

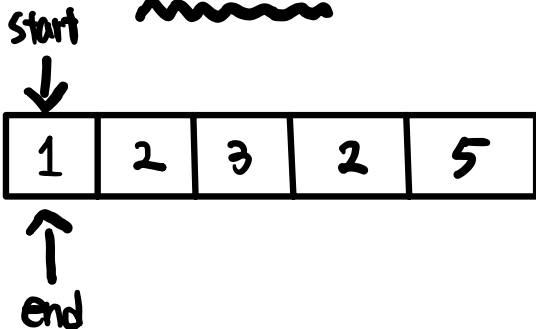
| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 2 | 5 |
|---|---|---|---|---|

two pointer algorithm

- 부분 연속 수열의 start, end의 위치를 기록한다.
- 특정 부분합을 M이라고 할 때, 구체적인 알고리즘은 다음과 같다.

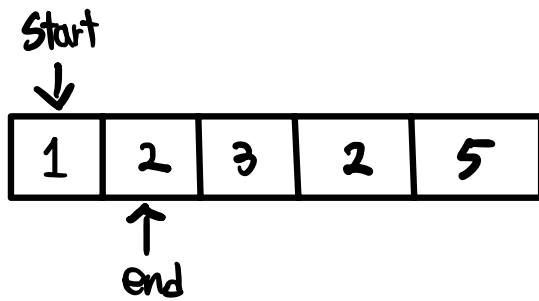
- 1) start & end가 첫 번째 원소의 index를 가르키도록 한다.
- 2) 현재 부분합이 M과 같다면, 카운트(결과)를 '+1' 한다.
- 3) 현재 부분합이 M보다 작다면, end를 1 증가시킨다.
- 4) 현재 부분합이 M보다 크거나 같다면, start를 1 증가시킨다. (이때, start 1 증가시킨 arr[start] 값을 부분합에서 뺀다.)
- 5) 모든 case를 확인할 때 까지 2)~4) 경로를 반복한다.

[step 0] "이때, M=5"



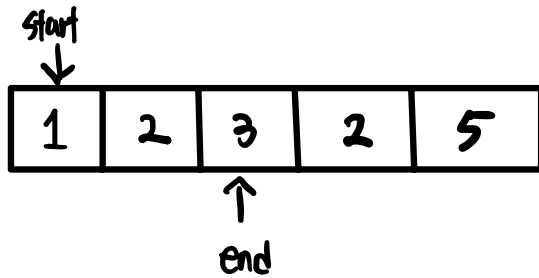
⇒ 부분합이 1 이므로 end를 증가시킨다. (+1)

[step 1]



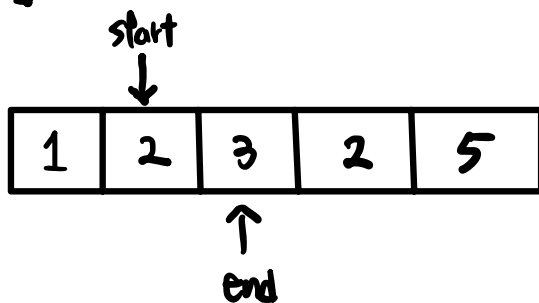
⇒ 부분합은 $3 < 5$ 이므로 $end += 1$

[step 2]



⇒ 부분합은 $6 > 5$ 이므로 $start += 1$ 하고,
부분합 $-= arr[이전 start]$ 해준다.

[step 3]



⇒ 부분합은 $5 == 5$ 이므로 $count += 1$
해주고 $start += 1$ 해주며 부분합 $-= arr[이전 start]$
해준다.

⋮

결과적으로 $\{(2,3), (3,2), (5)\}$ 라는 결과를 얻을 수 있다.