

[Algorithm] 39강 : 투 포인터(Two Pointers) 알고리즘의 정의와 구현 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-12-01 오전 8:55

수정한 날짜: 2020-12-01 오전 8:56

URL: <https://continuous-development.tistory.com/206>

Algorithm

[Algorithm] 39강 : 투 포인터(Two Pointers) 알고리즘의 정의와 구현

2020. 12. 1. 08:55 수정 삭제 공개

Two Pointers(투 포인터)

1.1 투포인터 알고리즘

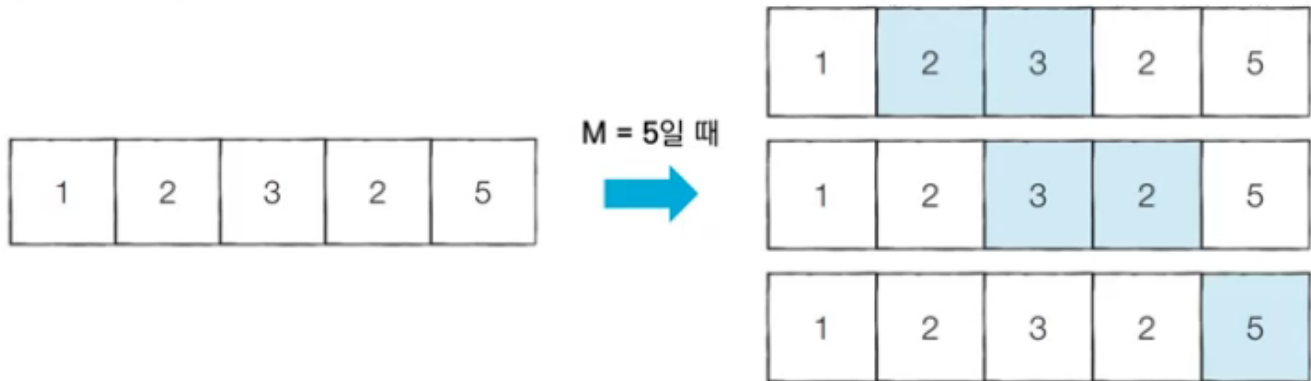
투 포인터 알고리즘은 리스트에 순차적으로 접근해야 할 때 두 개의 점의 위치를 기록하면서 처리하는 알고리즘을 의미한다

흔히 2,3,4,5,6,7번 학생을 지목해야 할 때 간단히 '2번부터 7번까지의 학생'이라고 부른다.

리스트에 담긴 데이터에 순차적으로 접근해야 할 때는 시작점과 끝점 2개의 점으로 접근할 데이터의 범위를 표현한다.

2.1 특정한 합을 가지는 부분 연속 수열 찾기(대표적인 문제)

- N개의 자연수로 구성된 수열이 있습니다.
- 합이 M인 부분 연속 수열의 개수를 구해보세요.
- 수행 시간 제한은 $O(N)$ 입니다.

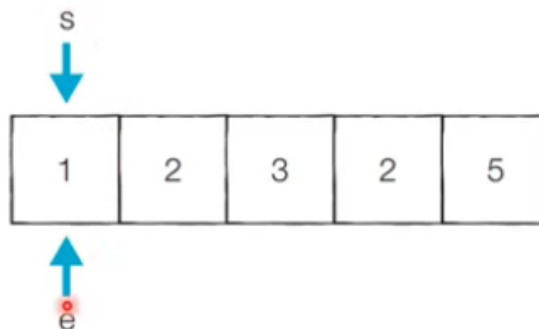


2.2 문제해결 아이디어

두 포인터를 활용하여 다음과 같은 알고리즘으로 문제를 해결할 수 있다.

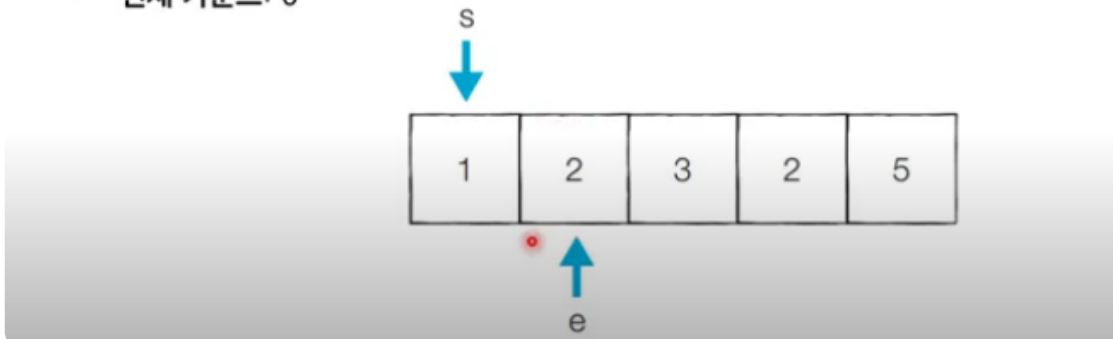
1. 시작점과 끝점이 첫 번째 원소의 인덱스(0)를 가리키도록 한다.
2. 현재 부분 합이 M과 같다면, 카운트 한다.
3. 현재 부분 합이 M보다 작다면, end를 1 증가시킨다.
4. 현재 부분 합이 M보다 크거나 같다면, start를 1 증가시킨다.
5. 모든 경우를 확인할 때까지 2번부터 4번까지의 과정을 반복한다.

- **M = 5**
- [초기 단계] 시작점과 끝점이 첫 번째 원소의 인덱스를 가리키도록 합니다.
 - 현재의 부분합은 1이므로 무시합니다.
 - 현재 카운트: 0



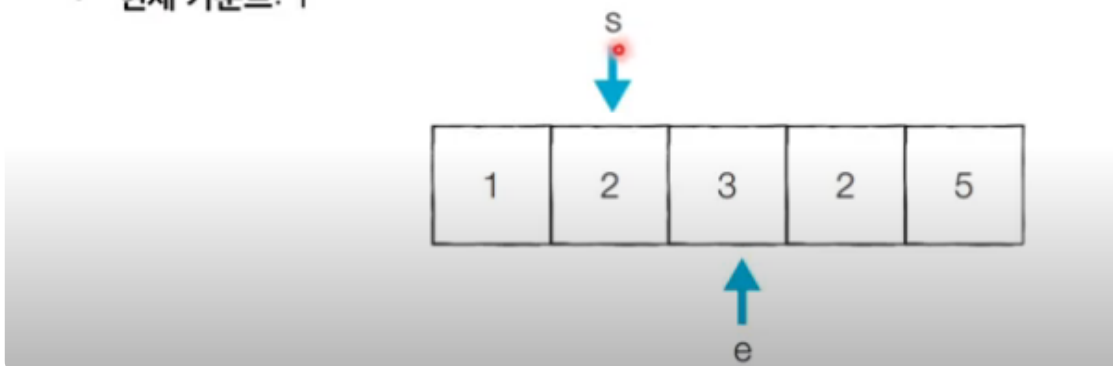
- $M = 5$

- [Step 1] 이전 단계에서의 부분합이 1이었기 때문에 end를 1 증가시킵니다.
 - 현재의 부분합은 3이므로 무시합니다.
 - 현재 카운트: 0



- $M = 5$

- [Step 3] 이전 단계에서의 부분합이 6이었기 때문에 start를 1 증가시킵니다.
 - 현재의 부분합은 5이므로 카운트를 증가시킵니다.
 - 현재 카운트: 1



이런식으로 합이 적으면 end index를 합이 크면 start index를 늘린다.

2.3 특정한 합을 가지는 부분 연속 수열 찾기

```
n = 5 # 데이터의 개수 N
m = 5 # 찾고자 하는 부분합 M
data = [1,2,3,2,5] # 전체 수열

count = 0
interval_sum = 0
end = 0

# start를 차례대로 증가시키며 반복
for start in range(n):
```

```

# end를 가능한 만큼 이동 시키기
while interval_sum < m and end < n:
    interval_sum += data[end]
    end += 1
# 부분합이 M일 때 카운트 증가
if interval_sum == m:
    count += 1
    interval_sum -= data[start]

print(count)

```

'Algorithm' 카테고리의 다른 글

[Algorithm] 38강 : 투 포인터(Two Pointers) 알고리즘의 정의와 구현

[Algorithm] 38강 : 투 포인터(Two Pointers) 알고리즘의 정의와 구현

[Algorithm] 38강 : 에라토스테네스의 체 알고리즘의 정의와 구현

[Algorithm] 37강 : 소수 판별 알고리즘의 정의와 구현

[Algorithm] 36강 : 위상정렬 알고리즘의 정의와 구현

[Algorithm] 35강 : 크루스칼 알고리즘의 정의와 구현

two point

투 포인터

투 포인터 구현

투 포인터 정의



나아무늘보

혼자 끄적끄적하는 블로그 입니다.

