



Two Pointer

투 포인터 알고리즘

- 두개의 점 (포인터, 인덱스를 가리키는 변수)를 조작해가며 문자열, 배열 등의 순차적인 데이터 구조를 선형적으로 탐색하는 알고리즘
- 주로 정렬된 배열, 리스트에서 두 항목의 공통된 항목을 찾는 문제에 활용
- 완전 탐색 풀이 대신 이 알고리즘을 활용하면 보다 메모리와 시간 효율성 높일 수 있음
각 배열 끝에 다다르는데 $O(N)$ 이기에 이를 모두 합쳐도 $O(N)$ 의 시간 복잡도를 가짐
(배열의 길이에 선형적으로 비례하는 시간소요)

투 포인터 알고리즘 - 특정한 합 갖는 부분 연속 수열

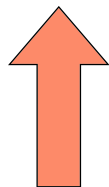
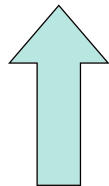
- 8개의 자연수로 구성된 수열이 있다 ($N = 8$)
- 부분 연속 수열의 합이 5인 수열의 개수를 구하라 ($M = 5$)

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

투 포인터 알고리즘 - 특정한 합 갖는 부분 연속 수열

- 8개의 자연수로 구성된 수열이 있다 ($N = 8$)
- 부분 연속 수열의 합이 5인 수열의 개수를 구하라 ($M = 5$)

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

- 시작점  $start = data[0]$, 수열합이 5보다 클 경우 우측 이동 (for 반복문을 통해 한칸씩 이동)
- 끝점  $end = data[0]$, 수열 합이 5보다 작을 경우 우측 이동(for 반복문 내부 while 반복문)
- 수열 합 시작점과 끝점을 모두 포함하는 범위 내 값의 총 합, 수열 합이 5일 경우 $cnt += 1$ 진행

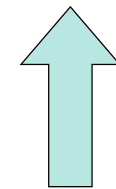
투 포인터 알고리즘 - 특정한 합 갖는 부분 연속 수열

- 8개의 자연수로 구성된 수열이 있다 ($N = 8$)
- 부분 연속 수열의 합이 5인 수열의 개수를 구하라 ($M = 5$)

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

시작점

끝점



최종적으로 두 점이 모두 배열의 끝에 도달하면 종료

투 포인터 알고리즘 - 특정한 합 갖는 부분 연속 수열

Python ▾

복사 캡션 ...

```
n = 8
m = 5
data = [3, 2, 4, 1, 2, 2, 1, 5]

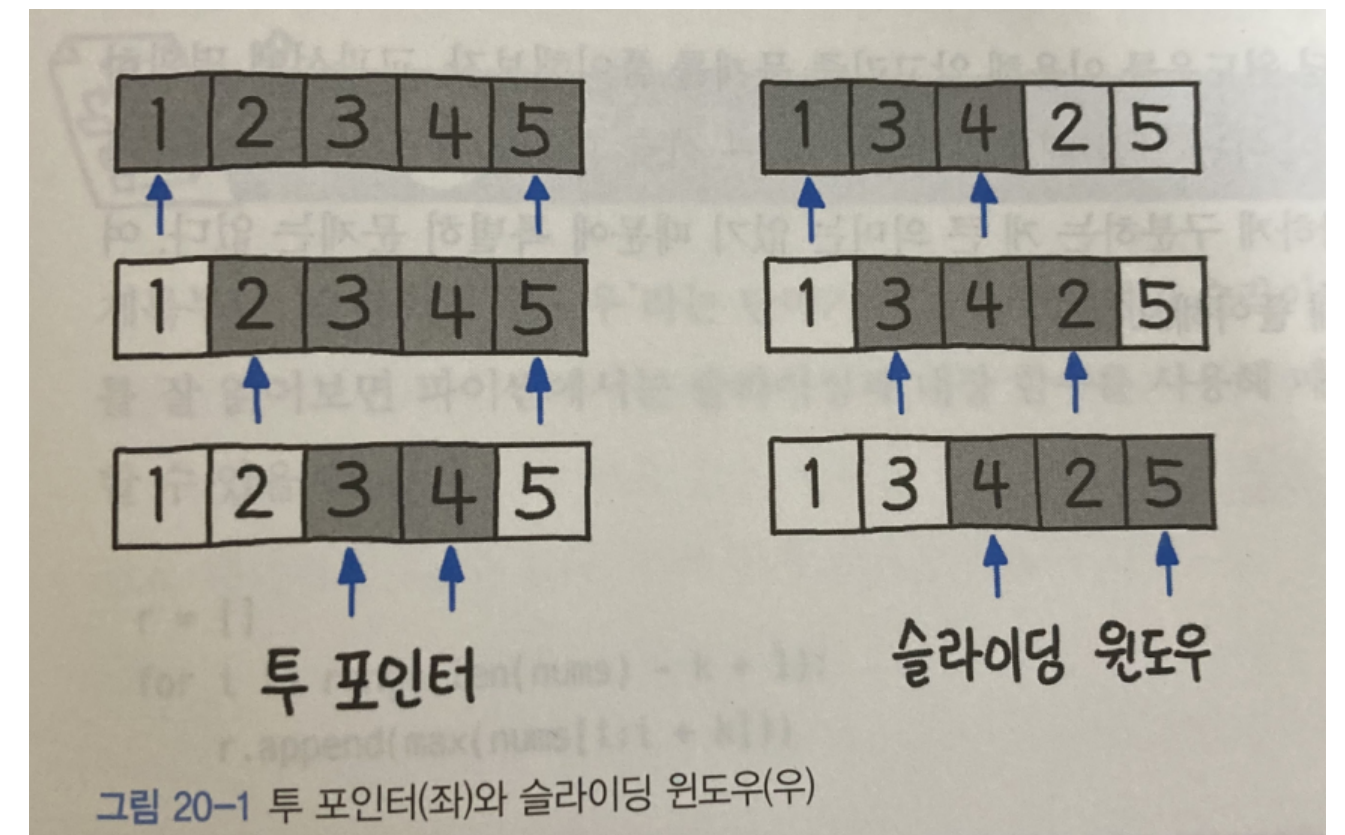
cnt = 0
interval_sum = 0
end = 0

for start in range(n):
    # 끝점을 이동시키는 상황 = 수열합이 m보다 작을 때 (끝점은 index n-1까지만 이동 가능)
    # while 문 종료 시점은 합이 같거나 클때 (+끝점이 배열 끝에 도달시)
    while interval_sum < m and end < n:
        interval_sum += data[end]
        end += 1
    # 수열 합이 m과 일치할 때 cnt 개수 증가
    if interval_sum == m:
        cnt += 1
    # 다음 시작점으로 시작하기 전, 이전 시작점의 위치값 삭제해서 수열합 맞춰놓기
    interval_sum -= data[start]

print(cnt)
```

투 포인터 VS 슬라이딩 윈도우

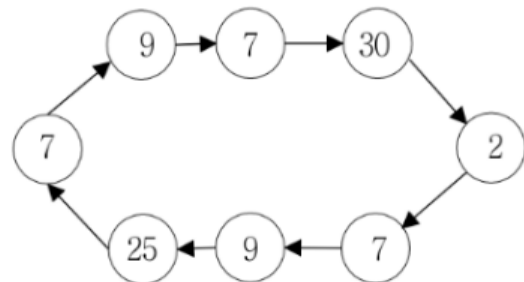
- 두 알고리즘 모두 배열과 같은 순차적인 데이터 구조에서 활용
- 슬라이딩 윈도우는 고정된 크기의 윈도우 (부분 배열, 부분 리스트)를 유지하며 배열 탐색 교집합의 정보를 공유하고, 차이가 나는 양 끝 원소만 갱신하는 방법
- 연속적인 구간의 최대 최소값 등의 문에 사용
일반적으로 $O(N)$ 또는 $O(N\log N)$ 의 시간 소요
- 투포인터 알고리즘과 연동하여 많이 사용



2531_회전 초밥

문제

회전 초밥 음식점에는 회전하는 벨트 위에 여러 가지 종류의 초밥이 접시에 담겨 놓여 있고, 손님은 이 중에서 자기가 좋아하는 초밥을 골라서 먹는다. 초밥의 종류를 번호로 표현할 때, 다음 그림은 회전 초밥 음식점의 벨트 상태의 예를 보여주고 있다. 벨트 위에는 같은 종류의 초밥이 둘 이상 있을 수 있다.



새로 문을 연 회전 초밥 음식점이 불경기로 영업이 어려워져, 다음과 같이 두 가지 행사를 통해서 매출을 올리고자 한다.

1. 원래 회전 초밥은 손님이 마음대로 초밥을 고르고, 먹은 초밥만큼 식대를 계산하지만, 벨트의 임의의 한 위치부터 k 개의 접시를 연속해서 먹을 경우 할인된 정액 가격으로 제공한다.
2. 각 고객에게 초밥의 종류 하나가 쓰인 쿠폰을 발행하고, 1번 행사에 참가할 경우 이 쿠폰에 적혀진 종류의 초밥 하나를 추가로 무료로 제공한다. 만약 이 번호에 적혀진 초밥이 현재 벨트 위에 없을 경우, 요리사가 새로 만들어 손님에게 제공한다.

위 할인 행사에 참여하여 가능한 한 다양한 종류의 초밥을 먹으려고 한다. 위 그림의 예를 가지고 생각해보자. $k=4$ 이고, 30번 초밥을 쿠폰으로 받았다고 가정하자. 쿠폰을 고려하지 않으면 4가지 다른 초밥을 먹을 수 있는 경우는 (9, 7, 30, 2), (30, 2, 7, 9), (2, 7, 9, 25) 세 가지 경우가 있는데, 30번 초밥을 추가로 쿠폰으로 먹을 수 있으므로 (2, 7, 9, 25)를 고르면 5가지 종류의 초밥을 먹을 수 있다.

회전 초밥 음식점의 벨트 상태, 메뉴에 있는 초밥의 가짓수, 연속해서 먹는 접시의 개수, 쿠폰 번호가 주어졌을 때, 손님이 먹을 수 있는 초밥 가짓수의 최댓값을 구하는 프로그램을 작성하시오.

2531_회전 초밥

입력

첫 번째 줄에는 회전 초밥 벨트에 놓인 접시의 수 N , 초밥의 가짓수 d , 연속해서 먹는 접시의 수 k , 쿠폰 번호 c 가 각각 하나의 빈 칸을 사이에 두고 주어진다. 단, $2 \leq N \leq 30,000$, $2 \leq d \leq 3,000$, $2 \leq k \leq 3,000$ ($k \leq N$), $1 \leq c \leq d$ 이다. 두 번째 줄부터 N 개의 줄에는 벨트의 한 위치부터 시작하여 회전 방향을 따라갈 때 초밥의 종류를 나타내는 1 이상 d 이하의 정수가 각 줄마다 하나씩 주어진다.

출력

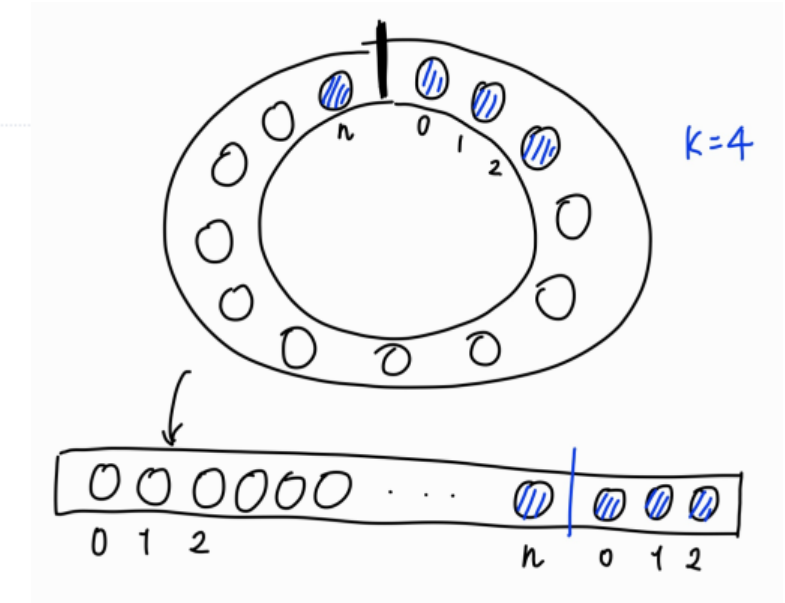
주어진 회전 초밥 벨트에서 먹을 수 있는 초밥의 가짓수의 최댓값을 하나의 정수로 출력한다.

```
n, d, k, c = map(int, input().split())
sushi = []
for _ in range(n):
    sushi.append(int(input()))
sushi = sushi + sushi[:k-1]
```

```
selected = set()
answer = 0
```

회전 초밥은 원형이기에
가장 마지막 초밥의 경우 처음 입력한 위치의 초밥부터 다시 시작

초밥 가짓수에 대한 최댓값이기에 set을 이용해서 가짓수 중복 제거
출력할 answer의 값을 0으로 초기 세팅



2531_회전 초밥

```
n, d, k, c = map(int, input().split())
sushi = []
for _ in range(n):
    sushi.append(int(input()))
sushi = sushi + sushi[:k-1]

selected = set()
answer = 0

for start in range(len(sushi)):
    end = start + k
    selected = set(sushi[start: end])
    s = len(selected)
    if c not in selected:
        s += 1
    if s > answer:
        answer = s
print(answer)
```

끝점은 시작점에 대해서 항상 고정적이기에
(슬라이딩 윈도우)

for문 내에서 시작점을 변화하면서
최대값일 경우 값 갱신

투 포인터 알고리즘 추천 문제들

- 11728. 배열 합치기 (정렬)
- 21921. 블로그 (슬라이딩 윈도우 누적합)
- 3273. 두 수의 합
- 2003. 수들의 합 2 (하늬님 픽)
- 22862. 가장 긴 짝수 연속한 부분 수열

- 2842. 집배원 한상덕 (DFS)
- [카카오인턴] 보석 쇼핑

<https://school.programmers.co.kr/learn/courses/13213/lessons/91426>



THANK