



자료구조와 알고리즘

4강 - 투 포인터 알고리즘

LECTURED BY SOONGU HONG



1. 투 포인터 핵심이론

* 투 포인터 알고리즘

- 길이가 N 인 배열에서 부분 배열 중 총합이 M 인 경우의 수를 구하는 문제에서 구간 합 알고리즘을 이용한다고 해도 N 의 값이 엄청 크다면 시간 초과가 일어납니다.
- 투 포인터 알고리즘은 이름 그대로 시작($start$)과 끝(end)을 가리키는 포인터 2개를 준비합니다.
- 처음에는 $start$ 와 end 포인터 모두 배열의 0번을 가리켜야 하며, 언제나 $start \leq end$ 를 만족해야 합니다.
- $start < N$ 인 동안 반복하며 다음과 같은 형태로 포인터를 움직입니다
 1. 현재 양 포인터 사이의 부분합이 M 보다 크면 $start++$
 2. 그렇지 않으면 $end++$



2. 핵심 예제 풀어보기

(백준 알고리즘 2018번)

<https://www.acmicpc.net/problem/2018>

수들의 합 5



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	32 MB	3875	1678	1404	50.631%

문제

어떠한 자연수 N 은, 몇 개의 연속된 자연수의 합으로 나타낼 수 있다. 당신은 어떤 자연수 $N(1 \leq N \leq 10,000,000)$ 에 대해서, 이 N 을 몇 개의 연속된 자연수의 합으로 나타내는 가지수를 알고 싶어한다. 이때, 사용하는 자연수는 N 이하여야 한다.

예를 들어, 15를 나타내는 방법은 15, 7+8, 4+5+6, 1+2+3+4+5의 4가지가 있다. 반면에 10을 나타내는 방법은 10, 1+2+3+4의 2가지가 있다.

N 을 입력받아 가지수를 출력하는 프로그램을 작성하시오.

입력

첫 줄에 정수 N 이 주어진다.

출력

입력된 자연수 N 을 몇 개의 연속된 자연수의 합으로 나타내는 가지수를 출력하시오

* 문제 분석

- 시간 제한이 2초이고 N 의 최대값이 1천만입니다.
- 이 문제는 단순히 구간합을 구하는 게 아니라 구간합의 경우의 수를 구해야 하기 때문에 투 포인터 알고리즘이 적합하다고 볼 수 있습니다.

* 예제 입출력 분석

예제 입력 1 복사

15

예제 출력 1 복사

4

- 이전에 분석한 투 포인터 이동 원칙에 따라 부분합 M 과 주어진 자연수 N 을 지속적으로 비교하면서 포인터를 이동시킵니다.
- M 이 N 과 같거나 작으면 `end`포인터 한 칸 이동 후 포인터가 가리키는 값 합산
- M 이 N 보다 크면 `start`포인터가 가리키는 값 차감 후 한 칸 이동

n = 15

se m=1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=3

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=6

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=10

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=15 , count=1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=21
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

 s e m=20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

 s e m=18
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

 s e m=15, count=2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

 s e m=22
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=18
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=13
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=21
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=15, count=3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=24
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=17

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

se m=9

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=19

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

se m=10

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=21

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

se m=11

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

... 종략

se m=14

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e m=29

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

se m=15, count=4

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

s e , 반복문 종료

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

* 의사코드 작성하기

N 변수 저장

1부터 N까지의 자연수 배열 초기화

사용 변수 초기화 (M=1, start=0, end=0, count=0)

While (end < N) {

 if (M == N) count증가, end증가, M += 배열[end]

 else if(M > N) M -= 배열[start], start값 증가

 else if(M < N) end증가, M += 배열[end]

}

Count 출력

```
Scanner sc = new Scanner(System.in);
int N = sc.nextInt();

int[] arr = new int[N];
for (int i = 0; i < arr.length; i++) {
    arr[i] = i + 1;
}

int start = 0, end = 0; // 시작, 끝 포인터
int M = 1; // 부분합 저장
int count = 0; //경우의 수 저장

while (end < N) {
    if (M == N) {
        count++; // 경우의 수 증가
        end++;
        if (end < N) M += arr[end];
    } else if (M > N) {
        M -= arr[start];
        start++;
    } else {
        end++;
        M += arr[end];
    }
    //System.out.printf("start: %d, end: %d, M: %d, count: %d\n", start, end, M, count);
}

System.out.println(count);
```



3. 실전 예제 풀어보기

(백준 알고리즘 1940번, 1253번)

<https://www.acmicpc.net/problem/1940>

<https://www.acmicpc.net/problem/1253>