고급 소프트웨어 실습

분반: 1분반

학번: 20181687

이름: 정지석

과제 8.

1. Sliding Window 실습 3의 효율적인 풀이를 위해서는 어떤 자료구조(예: stack, queue, array …)를 사용해야 되는지를 이유와 함께 제시하시오.

Array를 사용해야한다. 3번 문제를 해결하기 위해서는 여러 개의 숫자 입력을 window size에 맞게 잘라서 연속적으로 평균을 계산해야 하는데, 이 때 각 숫자를 따로 다로 받는 것보다 하나의 배열에 넣어서 관리하는 것이 훨씬 효율적이다. 배열을 사용할 경우 반복문을 사용하여 window 위치에 따른 평균값을 빠르게 구할 수 있음을 확인할 수 있다.

1. 앞에서 다루지 않은 문제 중에 Two pointers와 Sliding window를 사용했을 때 더 효율적으로 풀 수 있는 문제를 각각 하나씩 들고, 어떤 면에서 효율적인지 알고리즘을 설명하시오.

2-1. Two Pointers

하나 이상의 연속된 소수의 합으로 나타낼 수 있는 자연수들이 있다.

Ex) 2+3+5+7+11+13 = 11+13+17 = 41

7+13 = 20 과 같이 연속되지 않는 소수의 경우는 포함하지 않으며, 3+5+5+7 = 20과 같이 같은 소수가 여러 번 반복되는 경우도 포함하지 않는다.

어떤 자연수가 주어졌을 때, 이 자연수가 연속된 소수의 합으로 나타낼 수 있는 경우의 수를 구하시오.

해당 문제를 풀기위해서는 먼저 소수를 찾는 방법인 ‘에라토스테네스의 체’를 사용해야한다. 이렇게 해당 방법으로 소수의 전체 목록을 배열에 저장하여 구하면 이제 본격적으로 문제를 풀 수 있다.

이제 소수 배열을 반복문을 통해 돌면서 연속된 소수들을 더해주며 주어진 자연수와 일치하는 지를 찾아야 하는데, 이때 하나의 index 만으로 반복문을 돌게 되면 작업이 비효율적이게 된다. 연속된 소수 값을 더해주다가(sum) 그 값이 주어진 자연수를 넘어가면 제일 앞 쪽에 있는 값을 빼주어야 하는데, 이 작업은 하나의 index로는 하기 힘든 작업이다. 따라서 Two pointer를 사용하여 하나의 index는 소수들을 더해주는 용도로, 나머지 하나는 제일 앞의 소수를 빼주는 용도로 사용해야 한다. 이렇게 두개의 pointer를 사용해주면 해당 문제를 간단하게 해결할 수 있다.

2-2. Sliding Window

A씨는 N일 동안 웹사이트를 운영하고 있다. 최근 웹사이트의 방문자수가 늘어, A씨는 X일 동안 가장 많이 들어온 방문자 수와 그 기간들을 알고 싶어졌다. X일 동안 가장 많이 들어온 방문자 수와 기간이 몇 개 있는지를 구하시오.

해당 문제는 주어진 배열에서 고정된 개수 범위에서의 최대값과 그 최대값이 몇 번 나오는지를 구하는 문제이다. 여기서 주목해야 할 것은 ‘고정된 개수 범위’이다. 이는 위의 문제에서 ‘X일’에 해당하는 부분이고, 이 부분이 바로 ‘window’가 된다. 일정한 범위를 가지고 있는 window를 이동시키며 값을 구하는 문제는 sliding window 알고리즘을 사용하는 것이 가장 적절하다. 고정된 개수 범위를 지정하지 않고 위의 문제를 해결하려 하면 프로그램의 실행시간 만 길어지며 비효율적이게 된다.

따라서 위의 문제는 N의 길이의 배열을 X길이로 한 칸씩 이동하며 최댓값을 찾고, 그 최댓값들을 배열에 따로 저장한 뒤 그 값들 중 가장 숫자가 큰 것을 찾아 그 것의 개수를 구하는 식으로 해결하면 된다.