

고급 소프트웨어 실습

분반 : 4분반

학번 20191583

이름 : 김태곤

과제1. Sliding Window 실습 3의 효율적인 풀이를 위해서는 어떤 자료구조(예: stack, queue, array ...)를 사용해야 되는지를 이유와 함께 제시하시오. (1가지만 제시해도 ok)

Sliding Window 실습 3은 숫자 리스트와 윈도우 크기가 주어졌을 때, 윈도우를 오른쪽으로 이동시키며 평균값을 계산하는 문제이다. 간단하게 생각하면 처음 윈도우 크기만큼 숫자를 받아 평균값을 구한 후, 하나씩 옮기면서 먼저 입력 받은 숫자를 제거하고 추가 숫자를 포함시키면 된다. 이러한 방식에 가장 알맞은 자료구조는 queue이다.

Queue는 먼저 넣은 데이터가 먼저 나오는 FIFO 구조로 저장하는 형식이다. 파이썬 queue를 이용하여 이 문제를 해결하는 방법은 다음과 같다. 먼저 윈도우 크기만큼 queue.put을 이용해 데이터를 저장한다. 데이터를 넣으면서 sum에 데이터를 더해 합을 구한다. 이후 윈도우 크기부터 리스트 사이즈까지 반복문을 통해 리스트 안의 데이터에 접근한다. 그 과정에서 queue.get을 이용해 queue에 먼저 들어온 데이터를 제거하며 sum에 그 값을 제거한다. 그리고 반복문에서 얻은 리스트의 데이터를 다시 queue.put을 이용해 넣으면서 실습3을 효율적으로 해결할 수 있다.

이처럼 queue를 이용하면 데이터의 접근, 삽입, 삭제가 빠르다는 장점이 있다. 또한, 실습3처럼 중간 데이터의 접근이 불필요한 경우 사용하기에 더 유용하다는 장점이 있다.

과제2. 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.

회전초밥 문제는 회전 초밥 벨트에 N개의 접시가 있고, d가지 초밥 종류가 있을 때 연속해서 먹는 수k, 공짜 쿠폰 번호 c가 주어진다. k개의 서로 다른 연속된 초밥이 있을 때, 안에 공짜 쿠폰 번호인 c가 포함되어 있으면 연속된 초밥 하나를 더 먹을 수 있다. 이때 회전 초밥 벨트에서 먹을 수 있는 초밥의 가짓수의 최댓값을 구하는 문제이다. 이 문제는 Two pointers를 이용해 풀기 아주 적합한 문제이다. 먼저 시작할 때라면 k개 만큼 선택한다. 이때 중복이 있을 수 있으므로 dictionary로 저장한다. 이후 left와 right를 1씩 증가하면서 이동시킨다. Left dictionary의 값은 이동 전 1 감소시키고, 오른쪽은 1 증가시킨다. dictionary 값이 0이되면 set에 제거하고, 1이 되면 set에 추가한다. 이때 공짜 쿠폰이 현재 초밥 종류에 속해 있지 않다면 1을 더한 값으로 가짓수를 비교해 갱신한다. Left가 마지막에 도달할 때까지 진행하면 초밥의 최대 가짓수를 구할 수 있다. 이처럼 Two pointers를 이용하면 효율이 좋아진다. 만약 Two pointers를 이용하지

않았다면 완전 탐색을 적용해 두번의 반복문을 통해 $O(n^2)$ 의 시간 복잡도가 걸리게 된다. 하지만 Two pointer를 사용함으로써 $O(n)$ 의 시간 복잡도로 보다 빠르고, 효율적으로 해결할 수 있다.

$N \times N$ 배열에서 N 번째 큰 수 찾는 문제는 Sliding window를 이용해 풀면 효율이 좋다. 문제 해결 방법은 먼저 첫번째 줄을 입력 받아 배열에 저장하고 sort를 통해 정렬한다. 이후 $n-1$ 번 반복하며 줄 단위로 입력 받아 배열에 추가하고 정렬하여 n 번째까지 자르면 된다. 만약 이 문제를 단순하게 전부 입력 받은 뒤 정렬하면 많은 메모리를 잡아먹게 된다. 따라서 Sliding window를 이용해 해결함으로써 메모리를 아낄 수 있다.