CS202 Topic #1 Report

20170038
전산학부
권기훈

[apple]

N개의 사과 중 M개의 사과를 연속적으로 주었을 때의 최소값을 구하는 것이기 때문에 최솟값을 빠르게 찾을 수 있으며, 값만 알고 있으면 삭제 또한 용이한 자료 구조인 Set과 모든 케이스에 대하여 순차적으로 최솟값을 찾아야 하므로 FIFO의 Queue를 이용하는 것이 적합하다고 판단하였습니다. 이 때, Set에 중복된 값이 저장될 수 있으므로, 삽입한 순서와 Pair를 이뤄 대입하였습니다.

문제를 해결하는 과정에서는 초기 M개의 사과를 먼저 Queue와 Set에 넣어주었고, 앞의 M-1개의 사과는 끝쪽의 사과들과 다시 계산되어야 하기 때문에, Queue를 하나 더 만들어서 따로 저장하였습니다.

전체 사과를 순회하는 동안, 입력을 지속적으로 받으면서 Queue와 Set에 넣고 Queue에서 뺀 값을 Set에서 찾아 빼주면서 모든 경우에 대하여 Set에서 최솟값을 구하여 출력하는 방식으로 문제를 해결하였습니다.

[Shade]

전반적인 방향성은 사각형이 겹치는 x좌표를 기준으로 사각형들을 모두 쪼개서 단일의 직사각형들을 다시 더하는 방식입니다. 값을 입력 받으면 각 x값에 해당하는 y값의 범위와, 직사각형이 시작하는 부분인지, 끝나는 부분인지의 여부를 저장하는 pair를 만들어 vector에 저장하였습니다. 이 때 각각의 직사각형을 set에 저장하였습니다. Vector에 저장한 이유를 아래에 서술하겠습니다.

입력을 다 받은 이후에는 vector에 저장한 x좌표들을 차례로 순회하면서, 위에서 저장한 직사각형들에 의해서 y값이 바뀐다면 이를 추가해주었습니다. 이 과정에서 y값의 범위를 추가적으로 vector에 넣었습니다. 초기에는 set에 넣는 코드였고, for문을 도는 과정에서 항목이 추가되어 중복 순회가 생기는 것을 막기 위해 다른 set에 저장하는 방식을 택하였으나, 메모리가 부족해지는 것을 확인하였습니다. 따라서 vector 자료 구조로 바꾸어 사전에 지정한 부분까지만 순회를 시키고 뒤에 추가한 뒤, 정렬을 하는 방법으로 코드를 수정하였습니다.

그 이후에는 위의 정렬된 vector를 순회하면서 같은 x좌표끼리 묶어서 y값의 범위를 합쳐주었습니다. 예를 들어 1~3 2~4 인 경우 1~4로 만들어주는 것입니다. 이때 범위가 중복되지 않은 경우 합치지 않았습니다. 이렇게 묶은 x좌표들을 새 set에 저장하였습니다.

마지막으로 바로 위의 set을 순회하면서 직사각형의 시작 부분인 경우에만 dx를 구하고, dy값은 이미 저장한 y값의 차이 이므로, 이를 곱해서 모두 더해주었습니다. 이때 마지막 x좌표는 확인하기 위해서 마지막 값을 따로 저장한 뒤 비교하였습니다.