"뗏목 제작 (raft)" 문제 풀이

작성자: 조승현

부분문제 1

주어진 KOI 숲과 IOI 숲의 나무를 모두 사용하여 뗏목을 만들 때, 높이가 h인 직사각형의 가능한 최대 크기는 KOI 숲에서 높이 h인 연속한 나무의 최대 갯수 $f_1(h)$와 IOI 숲에서 연속한 나무의 최대 갯수 $f_2(h)$에 대해 $h \times (f_1(h) + f_2(h))$가 된다. 여기서 h는 나무들 중 적어도 하나의 높이와 일치해야 의미가 있으므로, 스택을 사용하여 히스토그램의 maximal한 직사각형을 구하는 방법을 이용하면 문제를 $O(N + M)$ 시간에 해결할 수 있다. 쿼리마다 이를 반복하면 $O(Q(N + M))$이 된다.

부분문제 2

위 방법으로 쿼리마다 $O(N + M)$시간에 문제를 해결하면 충분하다.

부분문제 4

높이가 50 이하이므로, 각 높이 h에 대해 IOI 숲에서 높이가 h인 연속한구간은 겹치지 않는 구간들이다. 세그먼트 트리를 이용하면 각 높이 h에 대한 최댓값을 $O(\log N)$ 시간에 구할 수 있으므로 $O(50 \times Q \log N)$ 시간에 문제를 해결할 수 있다.

이외의 부분문제를 해결하기 위해서는 다른 아이디어가 필요하다: KOI 숲에서 높이 h_1, 너비 l_1인 직사각형과 IOI 숲에서 높이 h_2, 너비 l_2인 직사각형이 존재한다면 이를 이용해 안정성이 $\min(h_1, h_2) \times (l_1 + l_2)$인 뗏목을 만들 수 있다. 또한, $(\infty, 0)$도 각 숲의 (h, l)쌍에 포함시킨다면, 최대 안정성은 앞서 말한 경우를 통해서 나옴을 증명할 수 있다.

이제 다음과 같은 자료구조를 생각하자: (h_2, l_2)이 주어졌을 때 KOI 숲에서 (h_1, l_1)을 잘 선택하여 얻을 수 있는 $\min(h_1, h_2) \times (l_1 + l_2)$의 최댓값을 리턴

CHT와 세그먼트 트리를 이용하면 $O(N \log N)$ 시간의 초기화를 통해 $O(\log N)$시간에 (h_2, l_2) 쿼리를 처리하는 위 자료구조를 구현할 수 있다.

부분문제 3

IOI 숲에 해당하는 히스토그램에서 maximal한 직사각형을 모두 저장한 상태를 생각해보자. 문제 조건에 따라 이 직사각형들은 $[L[i], R[i]]$ 구간에 포함되거나, 완전히 포함하거나, 전혀 겹치지 않는다. 먼저 $[L[i], R[i]]$ 구간을 포함하는 경우는 (h_2, l_2) 쿼리 한번으로 해결할 수 있음을 알 수 있다. $[L[i], R[i]]$ 구간에 포함되는 경우는 모든 maximal한 직사각형 (h, l)에 대해 앞서 말한 자료구조에 쿼리의 리턴값을 저장해주면 2D 레인지 쿼리가 되어 $O(M \log M)$ 또는 $O(M \log^2 M)$ 시간에 문제를 해결할 수 있다.

부분문제 9

부분문제 3에서 처리되지 않은 부분은 maximal한 직사각형 (l, r, h)에 대해 $l \le L[i] \le r \le R[i]$ 또는 $L[i] \le l \le R[i] \le r$ 인 경우이다. 일반성을 잃지 않고 $L[i] \le l \le R[i] \le r$ 라 하자. 각 구간별로 maximal한 직사각형 (l, r, h)들을 저장하는 스택을 들고 있는 세그먼트 트리가 필요하다. maximal한 직사각형 (l, r, h)에 대해 $[l, r]$에 해당하는 $O(\log M)$개 구간에 대해, 스택에 해당 직사각형을 추가하자. $(L[i], R[i])$ 쿼리에 대해 $x = R[i]$일 때 (l, r, h) 직사각형을 통해 얻을 수 있는 최대 안정성을 $f(x)$라 하자. 그러면 각 직사각형에 대해

f의 그래프는 기울기가 증가하는 꺾은선 그래프이며, 한 x에 대해 $f(x)$의 값은 앞서 만든 자료구조의 쿼리 한번으로 얻을 수 있다.

각 노드에 대해 스택에 있는 직사각형들의 $f(x)$의 maximum에 해당하는 그래프를 생각했을 때, 각 직사각형이 최대인 부분은 연속된 구간으로 표현된다. 각 교차하는 지점을 계산해놓을 수 있고, 이를 통해 세그먼트 트리를 $O(M \log^2 M \log N)$시간에 관리할 수 있다. 그리고 각 $(L[i], R[i])$ 쿼리는 $O(\log M)$개 노드에 대해 쿼리 한번씩이면 충분하므로 $O(Q \log^2 M)$시간이면 충분하다. 따라서, 문제를 $O(M \log^2 M \log N + Q \log M \log N)$시간에 해결할 수 있다.