C5. Balloons

21600665 정현섭
21700477 윤다은

문제 분석

N개의 풍선에 대한 정보가 주어진다.
(풍선의 x좌표와 최대로 커질 수 있는 반지름 max_r)

각 풍선은 자신의 최대 반지름보다 커질 수 없고 다른 풍선에 닿으면 더이상 커질 수 없다.
첫번째 풍선부터 N번째 풍선까지 불 수 있는 최대로 불 것이다.
N번째 풍선까지 불었을 때, 각 풍선의 반지름을 구하여라.

문제풀이

(각 풍선의 x좌표와 반지름은 (x_i, r_i) 로 표현할 수 있다.)

수식 1) 풍선 A가 풍선 B와 접할 때 풍선 A의 반지름 $r_a = (x_a - x_b)^2 / (r_b * 4)$

<풀이 1>
풍선 A의 반지름 r_a를 구할 때, 기존의 모든 풍선들과 접하는 경우의 반지름들을 계산하여 그것들과 최대 반지름 중 작은 값을 r_a로 채택하는 것이다. (**수식 1** 이용)

이 풀이는 time complexity가 O(N^2)이 나와서 TLE가 되었다.
그래서 임의의 풍선 A의 반지름 r_a를 구할 때 기존의 풍선들과 비교하는 횟수를 줄여야 한다고 생각했다.

그 비교횟수를 줄이기 위해 아래의 두 가지 특성을 이용하였다.

특성 1) $x_p < x_a$ 이고 $r_a >= r_p$ 이면 p번째 풍선은 a번째 이후의 풍선과 접할 수 없다.
특성 2) $x_p < x_a$ 이고 $r_a <= r_p$ 이면 a번째 풍선은 p번째 풍선 이전의 풍선과 접할 수 없다.

응용(discussion)

1) 풍선을 부는 순서를 임의로 정할 수 있다면?
2) 풍선 중심의 y좌표가 주어진다면? → 같은 풀이로 풀 수 있을 것이다.

<풀이 2>
풍선 A의 반지름을 구할 때, 풍선 A이전의 모든 풍선에 대해 수식1을 적용하는 대신
stack을 만들어서 그 안에 들어있는 풍선들에 대해서만 수식1을 적용하였다.

[풍선 A의 반지름 r_a를 찾는 과정]
풍선 A의 반지름 r_a를 max_r로 두고,
stack이 empty가 되기 전까지 다음을 반복한다.

1) stack의 top에 있는 풍선(B)에 수식1을 적용한 값과 r_a 중 작은 값으로 r_a를 업데이트.
2) r_a와 r_b를 비교하여
   a) $r_a > r_b$ 이면 **특성 1**에 따라
      stack을 pop
   b) $r_a <= r_b$ 이면 **특성 2**에 따라
      지금의 r_a 값을 풍선 A의 최종 반지름으로 채택 후 반복을 멈춤.

이후 반지름이 결정된 풍선 A를 stack에 push.

위 과정을 1 ~ n 번째 풍선에 적용하여 반지름을 모두 구한다.

문제풀이 분석

- N : 풍선의 개수

시간복잡도 :
N개의 풍선에 대해 각 풍선의 반지름을 구하므로 O(N).
또한 stack에 들어있는 풍선을 pop하는 총 횟수는 N번을 넘지 못한다.
(stack에 push되는 풍선의 총 수는 N개 이므로.)

그러므로 최종적인 시간복잡도는 **O(N)**.

공간복잡도 :
풍선 개수 만큼의 저장공간이 필요하므로 **O(N)**.