

C8. Wells

1. Problem Analysis

- n 개의 마을의 x 좌표와 y 좌표가 주어질 때, 각 마을에서 d 안으로 $y=0$ 에 도달하기 위한 최소 우물의 개수 구하기.

$$n \leq 10000 \quad 0 \leq y \leq 10000 \quad y \leq d \quad -10000 \leq x \leq 10000$$

2. Solution

(마을과 $y=0$ 간의 거리) $\leq d$ 를 만족하는 x 값의 범위들간의 관계 파악

전체 흐름

- 1) 각 마을에서 $y=0$ 간의 거리 $\leq d$ 를 만족하는 x 값의 범위(x_1, x_2)를 계산하여 범위를 저장하는 배열 s_arr 에 저장한다. 저장한 배열은 x_1 의 값을 기준으로 합병 정렬한다.
- 2) s_arr 에서 0부터 $n-1$ 까지 $s_arr[i]$ 가 나머지 구간들과 겹치는지 검사한다. 만약 구간이 겹친다면 공동 우물을 쓸 수 있으므로, 우물의 수를 유지한다. 그렇지 않은 경우 우물의 수를 1증가시키고, 기준 구간을 해당 구간으로 업데이트한다.

범위 계산

조건1) $d == y$ (d 와 y 좌표가 같은 경우)

→ $y=0$ 과 한 점에서 만나므로 x 값의 범위는 마을의 x 좌표가 된다. $x_1 = x_2 = x$

조건2) d 와 마을의 y 좌표가 같지 않은 경우

→ 범위의 양 끝점은 해당 x 에서 부터 $\pm \sqrt{d^2 - y^2}$ 한 값이다 ($x_1 < x_2$).

겹침 계산

기준 구간의 x_2 의 값을 end 라 할 때, 초기 end 값은 첫번째 마을의 x_2 로 설정한다.

i 에 대해 ($1 \leq i \leq n-1$) 다음을 반복한다.

조건1) $s_arr[i]$ 의 $x_1 \leq end$ 인 경우

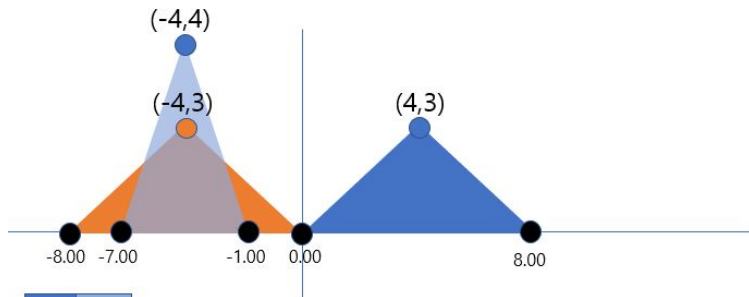
→ 구간 겹침이 일어나므로 우물 값을 증가시키지 않는다.

이때 $s_arr[i]$ 의 $x_2 < end$ 라면, end 를 x_2 로 업데이트한다.

조건2) $s_arr[i]$ 의 $x_1 > end$ 인 경우

→ 겹침이 더 이상 일어나지 않으므로 end 를 $s_arr[i]$ 의 x_2 로 업데이트 한 뒤, 우물의 수를 1 증가시킨다.

21701065 홍원표, 21700477 윤다은



-8 0

첫 pair. end = 0

-7 -1

end와 새로 들어온 $x_1(-7)$ 을 비교. $-7 < 0$. 우물 수 유지. (겹침 조건 1)
end와 새로 들어온 $x_2(-1)$ 을 비교. $-1 < 0$. end를 -1로 Update
end = -1

0 8

end와 새로 들어온 $x_1(0)$ 을 비교. $0 > -1$. 우물 수 +1. (겹침 조건 2)
end와 새로 들어온 $x_2(8)$ 을 비교. $8 > -1$. end를 8로 Update
end = 8

3. Solution Analysis (N : 전체 카드 개수)

1) 시간 복잡도 : $O(N \log(N))$

→ N번의 범위 계산

→ $N \log N$ 합병 정렬

→ N번의 겹침 계산

공간 복잡도 : $O(N)$

→ N개의 Pair

4. Discussion

전쟁 중, 2차원 좌표에 군사 초소가 여럿 배치되어 있고, 각 초소에서 이동할 수 있는 최대 거리가 d 이다. 각 초소는 아무 지휘센터나 도착할 수 있는 범위 안에 존재해야 할 때, $y=0$ 의 라인에 최소한의 지휘센터를 설치하는데 이 알고리즘을 이용할 수 있다.