

# 분할 정복 (도전)

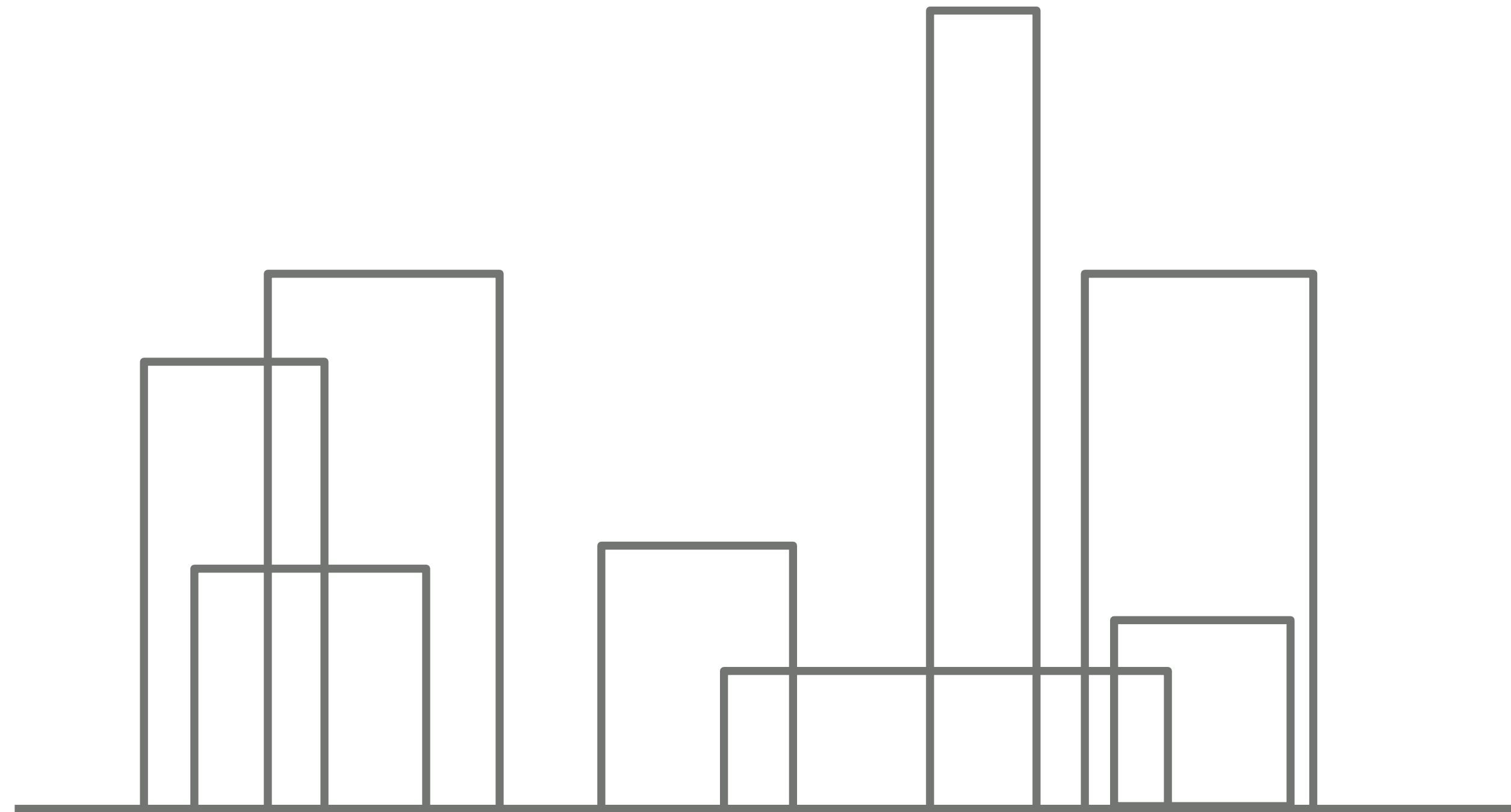
최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

# 스카이라인

<https://www.acmicpc.net/problem/1933>

- N개의 직사각형 건물이 있을 때 스카이라인을 구하는 문제
- $N \leq 100,000$



# 스카이라인

<https://www.acmicpc.net/problem/1933>

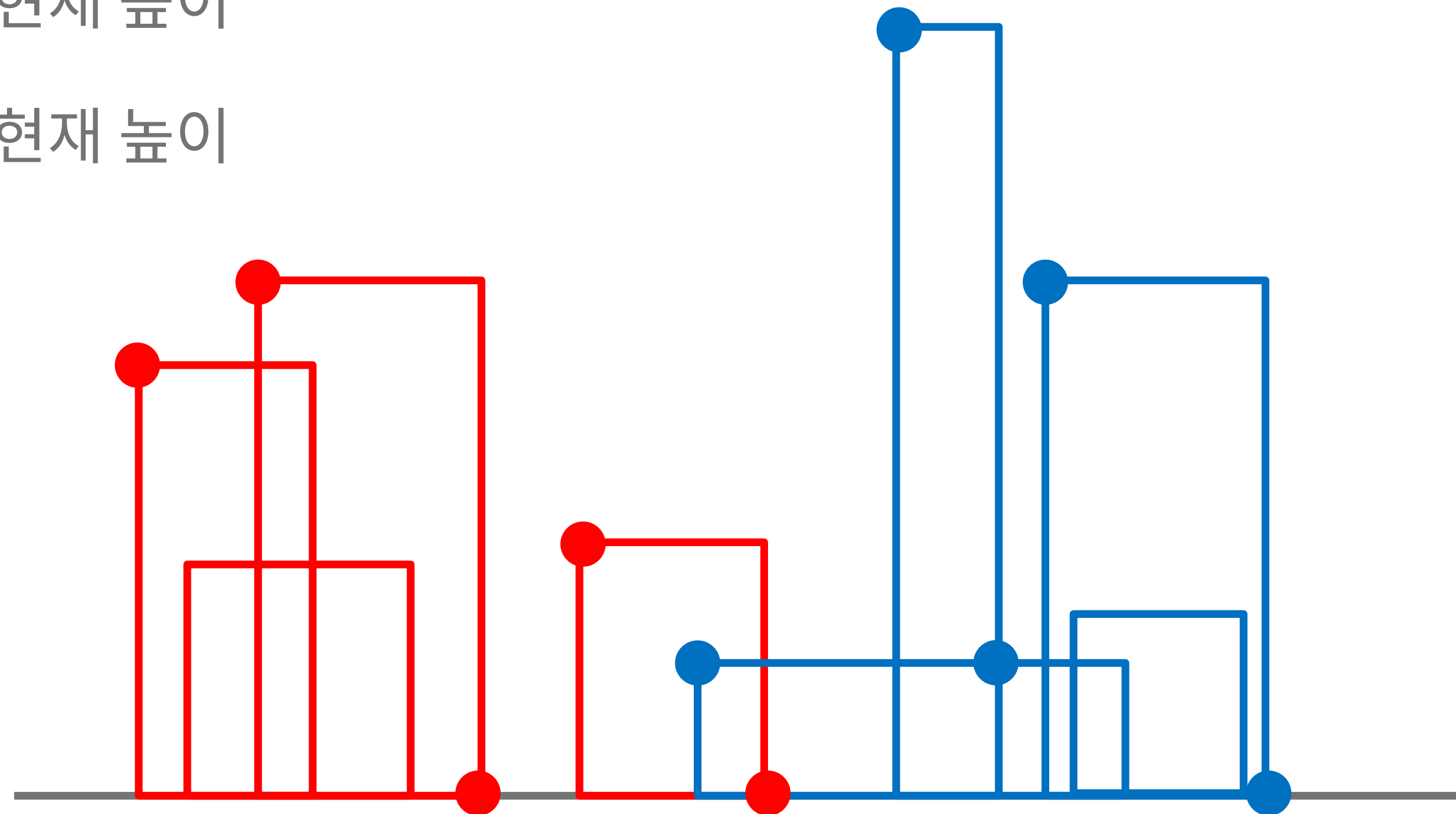
- N개의 직사각형 건물이 있을 때 스카이라인을 구하는 문제
- $N \leq 100,000$



# 스카이라인

<https://www.acmicpc.net/problem/1933>

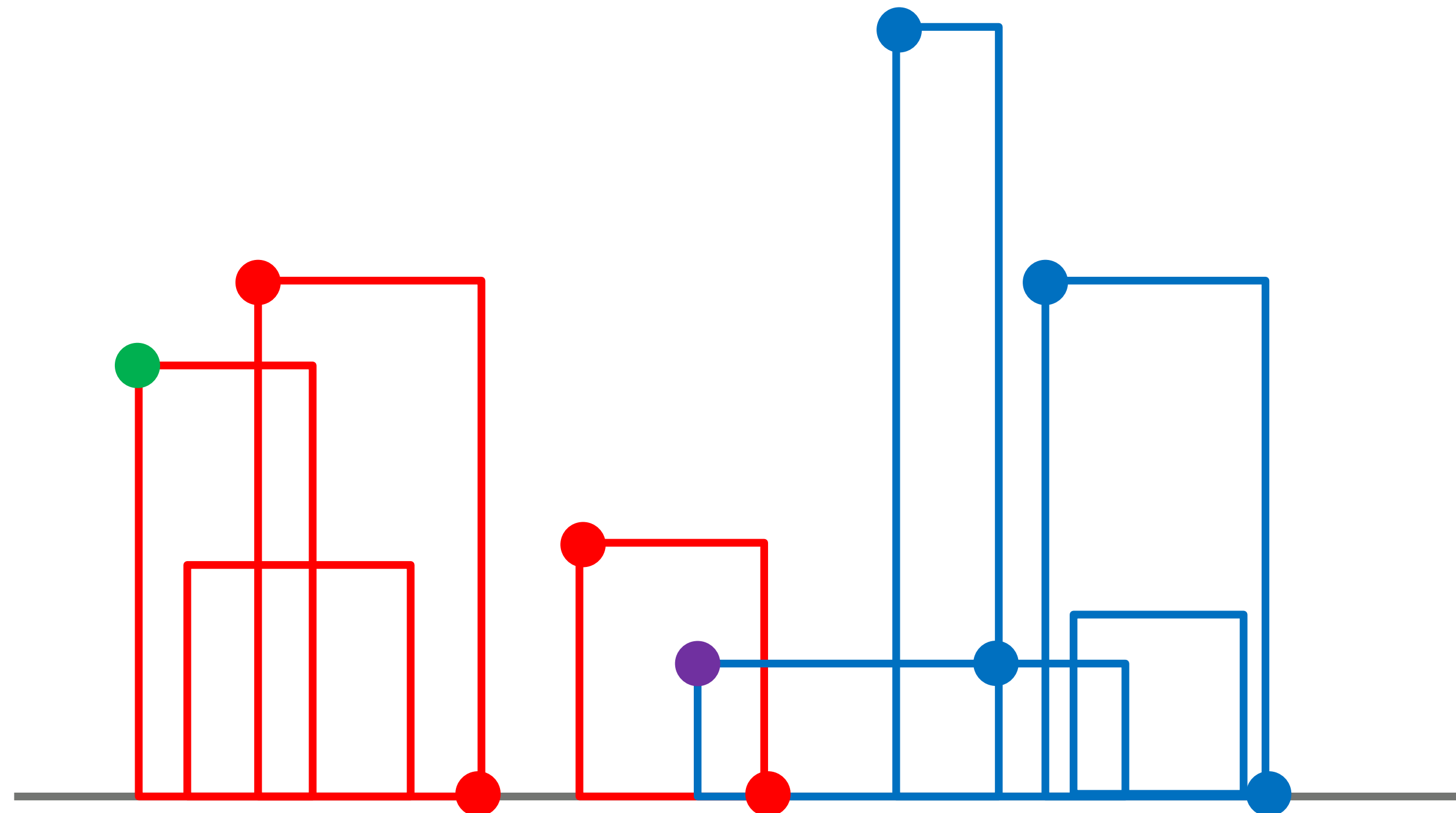
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)]
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)]
- h1 = 스카이라인 1의 현재 높이
- h2 = 스카이라인 2의 현재 높이



# 스카이라인

<https://www.acmicpc.net/problem/1933>

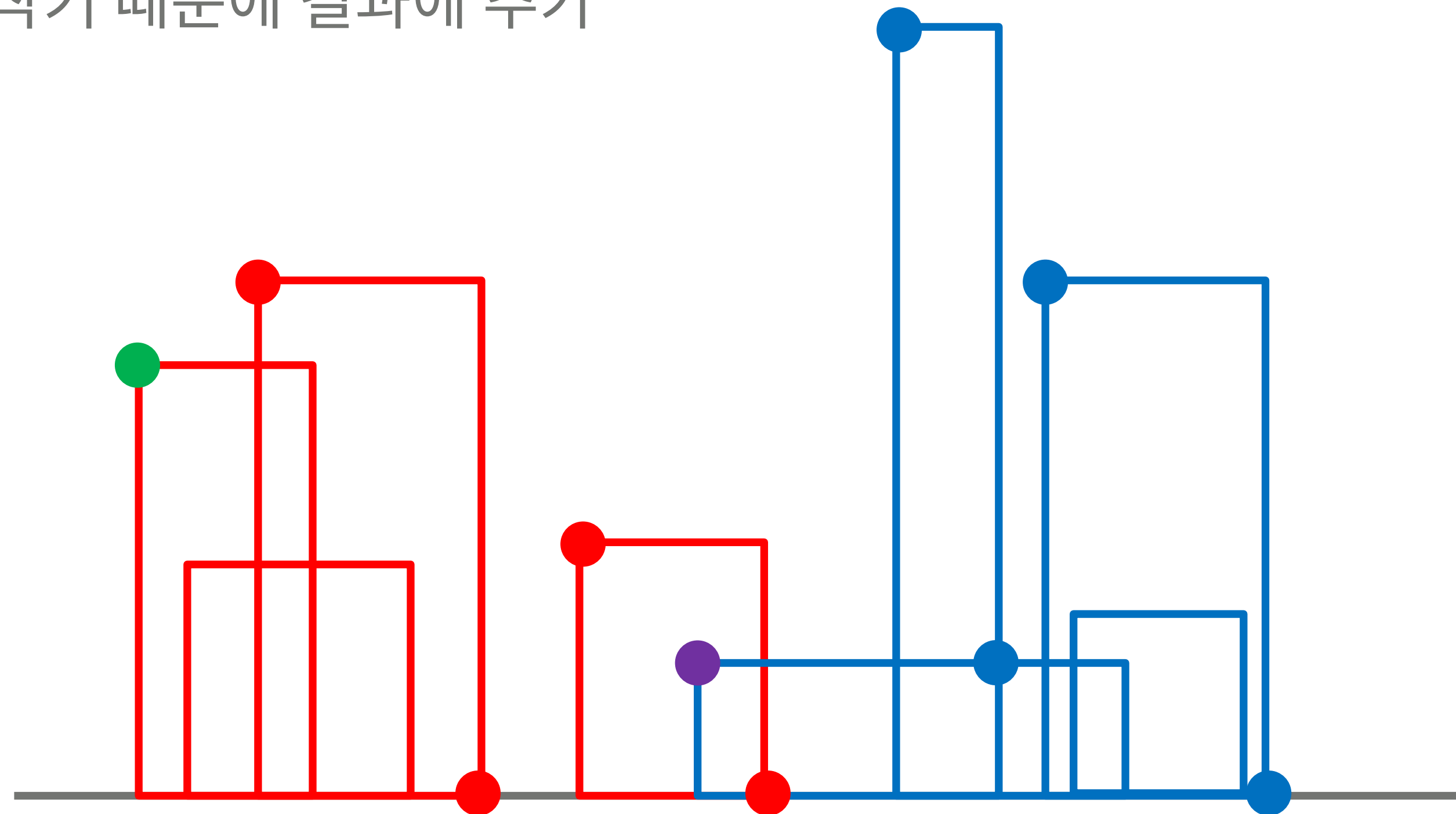
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 0
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 0
- 결과 = []



# 스카이라인

<https://www.acmicpc.net/problem/1933>

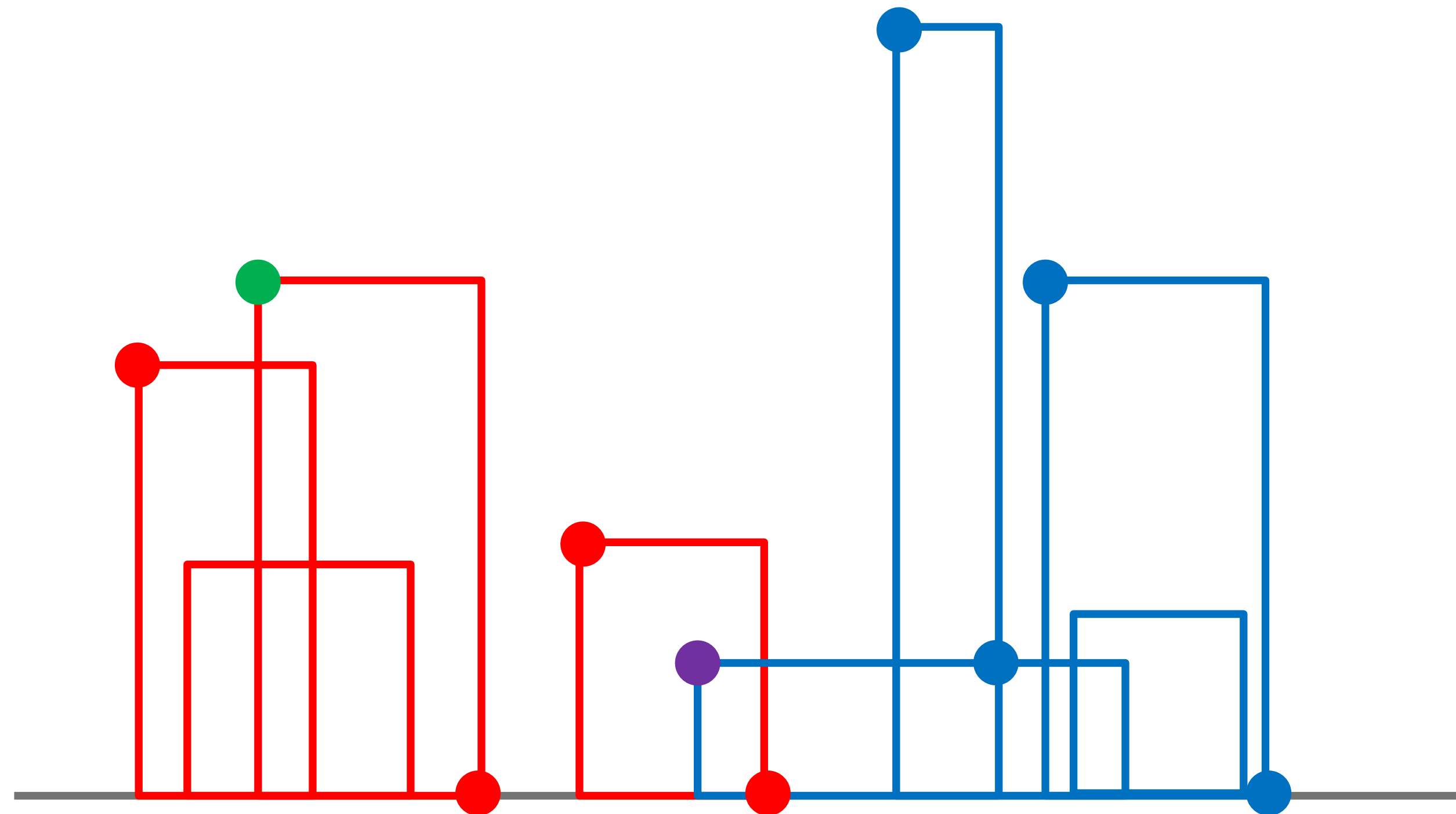
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)],  $h1 = 11$
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)],  $h2 = 0$
- (1, 11)의 x좌표가 더 작기 때문에 결과에 추가
- 결과 = [(1, 11)]



# 스카이라인

<https://www.acmicpc.net/problem/1933>

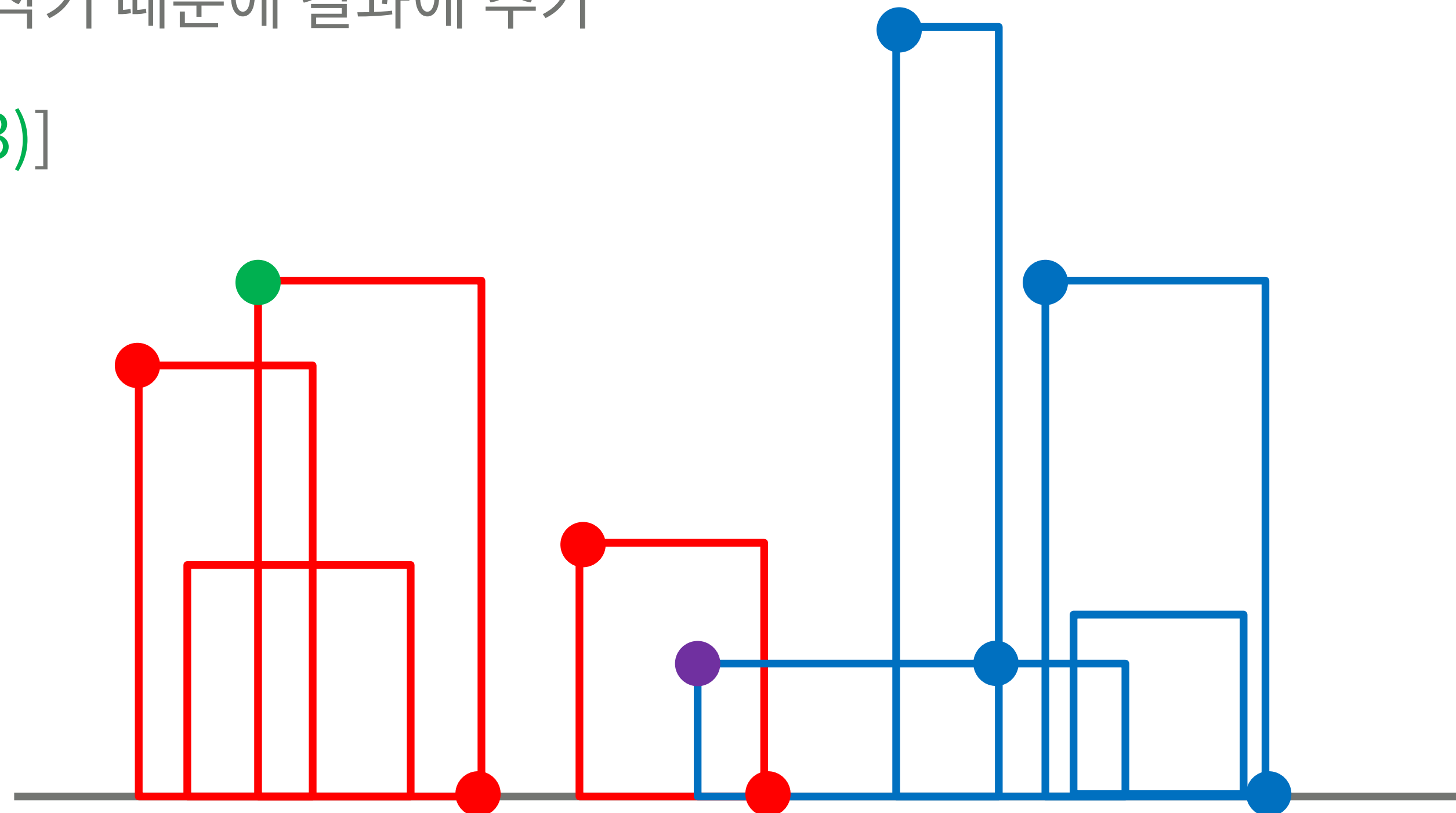
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 11
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 0
- 결과 = [(1, 11)]



# 스카이라인

<https://www.acmicpc.net/problem/1933>

- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 13
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 0
- (3, 13)의 x좌표가 더 작기 때문에 결과에 추가
- 결과 = [(1, 11), (3, 13)]

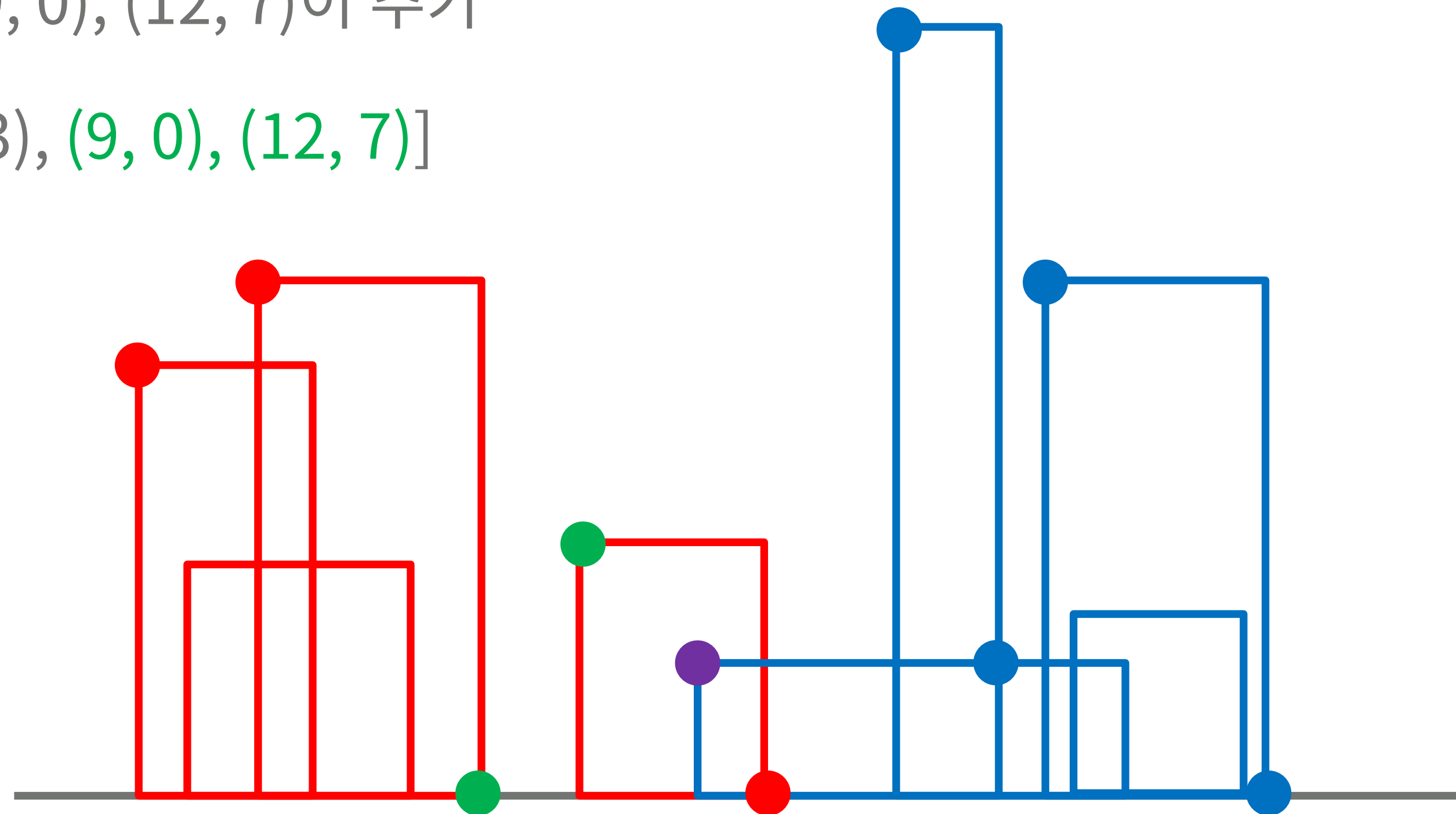




# 스카이라인

<https://www.acmicpc.net/problem/1933>

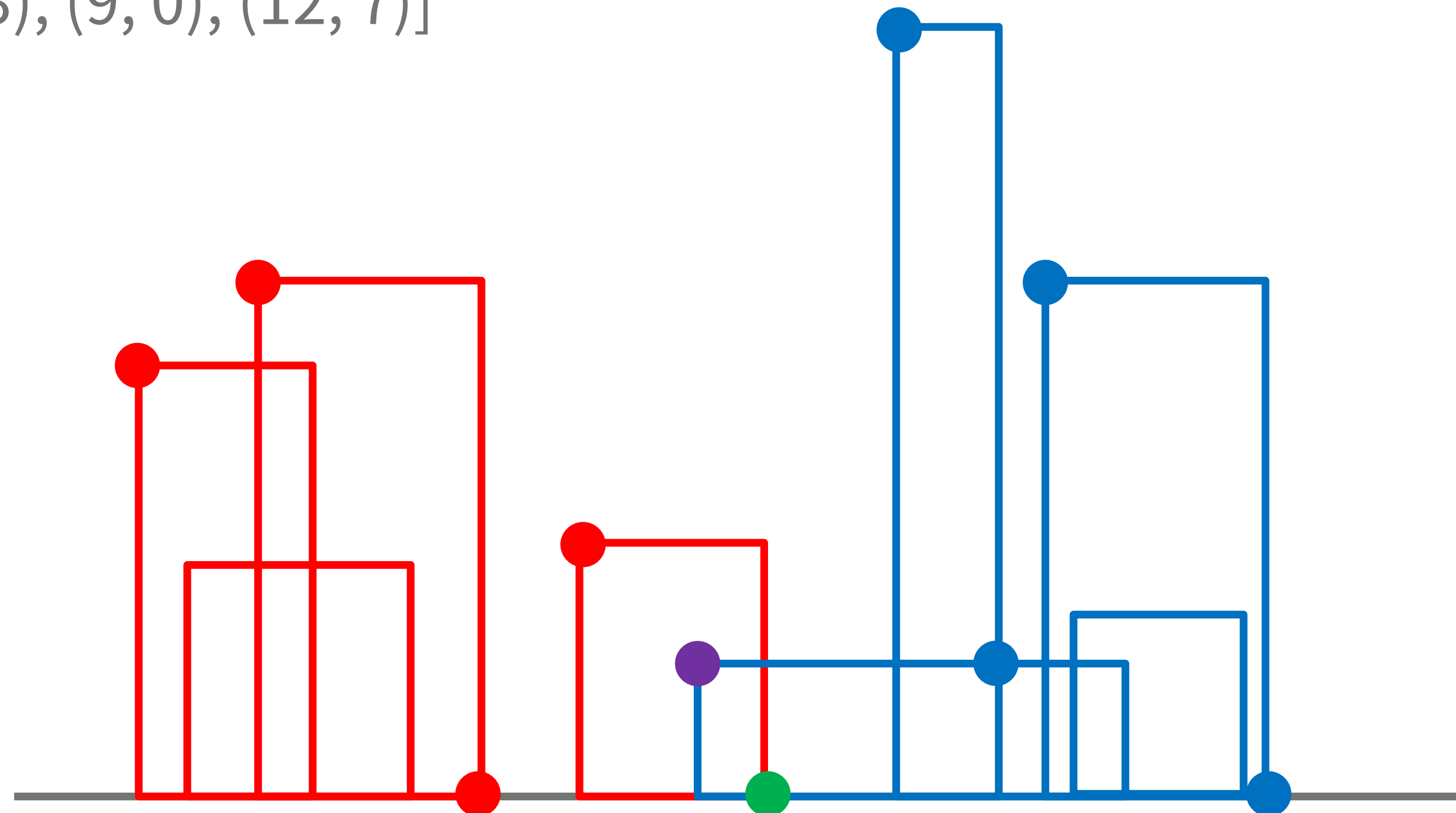
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)],  $h1 = 7$
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)],  $h2 = 0$
- 이와 같은 과정으로 (9, 0), (12, 7)이 추가
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7)]



# 스카이라인

<https://www.acmicpc.net/problem/1933>

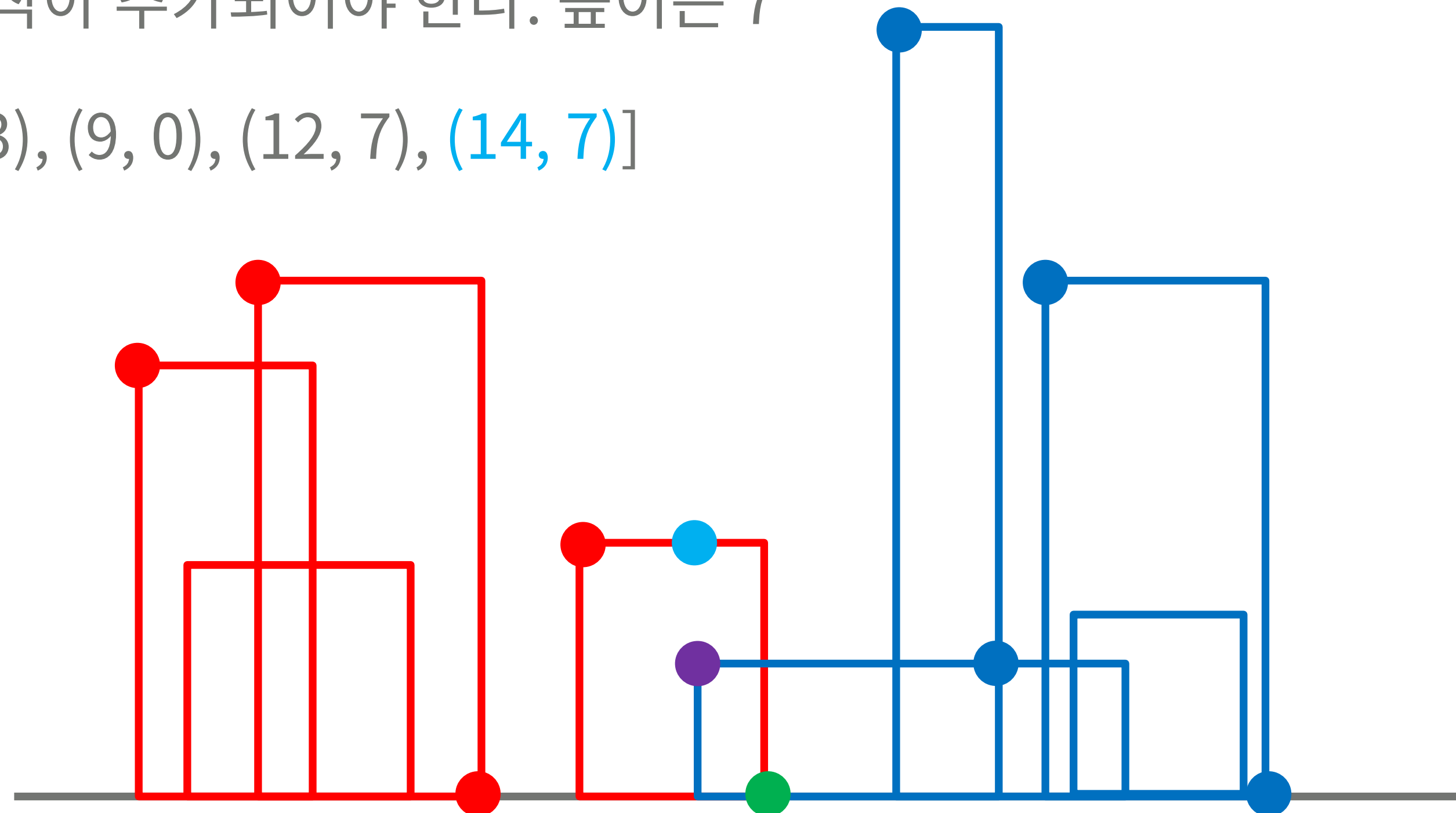
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 7
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 0
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7)]



# 스카이라인

<https://www.acmicpc.net/problem/1933>

- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 7
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 3
- (14, 3)의 x좌표가 더 작아 추가되어야 한다. 높이는 7
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7), (14, 7)]

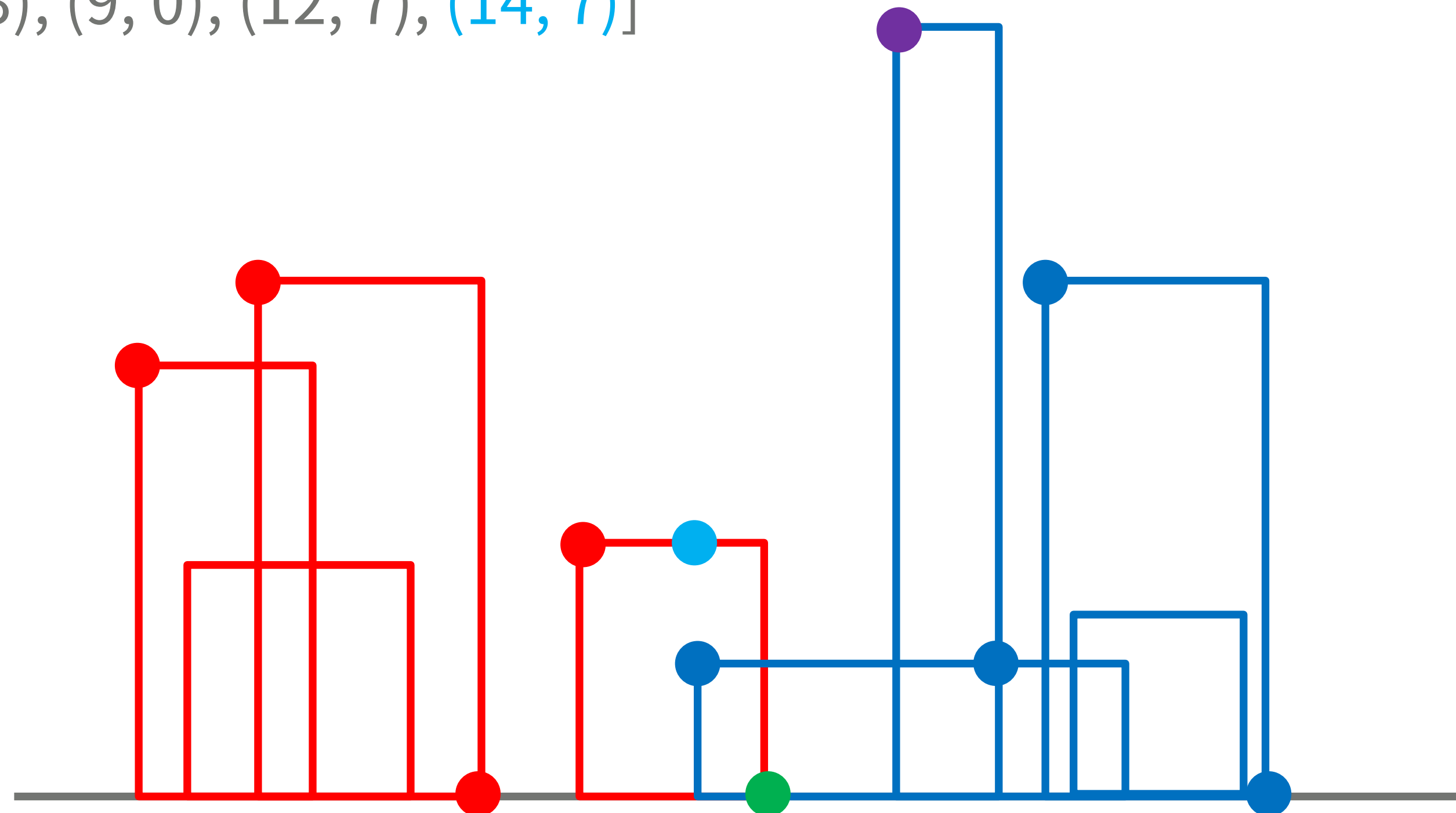


# 스카이라인

12

<https://www.acmicpc.net/problem/1933>

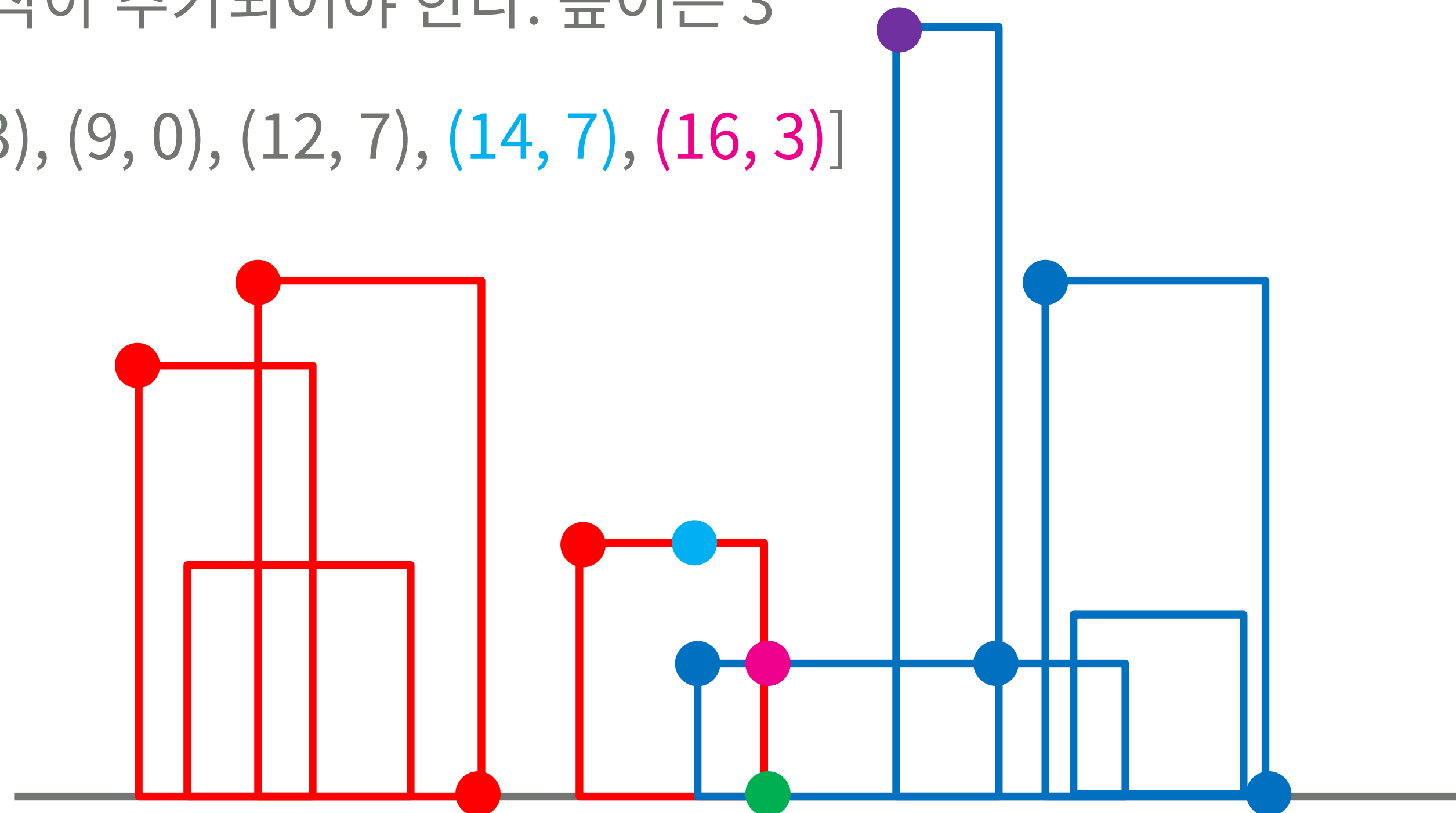
- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 7
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 3
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7), (14, 7)]



# 스카이라인

<https://www.acmicpc.net/problem/1933>

- 스카이라인 1 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 0)], h1 = 0
- 스카이라인 2 = [(14, 3), (19, 18), (22, 3), (23, 13), (29, 0)], h2 = 3
- (16, 0)의 x좌표가 더 작아 추가되어야 한다. 높이는 3
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7), (14, 7), (16, 3)]

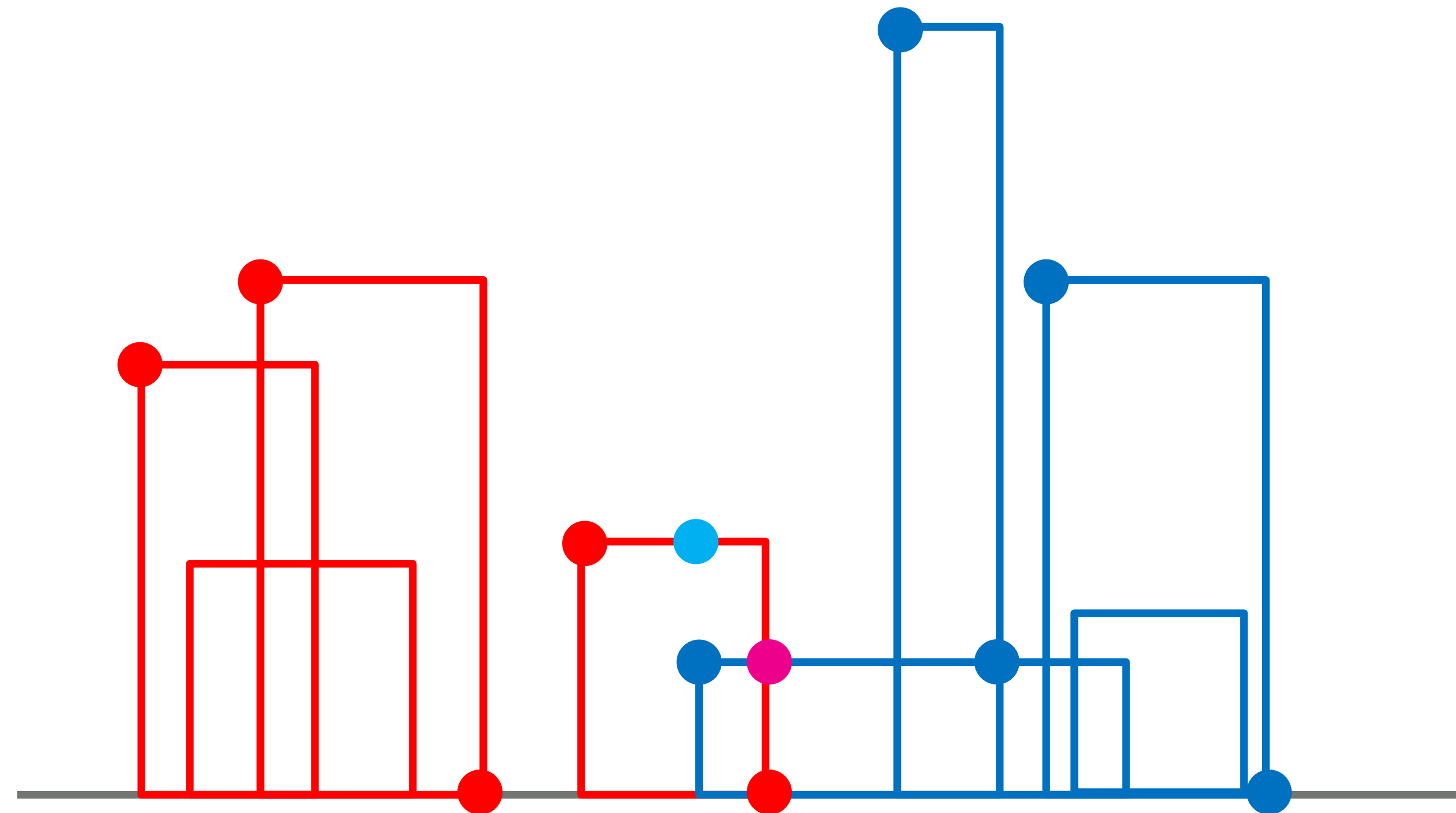


# 스카이라인

14

<https://www.acmicpc.net/problem/1933>

- 스카이라인 1에 더 이상 점이 없기 때문에 나머지 점을 모두 추가한다.
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7), (14, 7), (16, 3), (19, 18), (22, 3), (22, 13), (29, 0)]

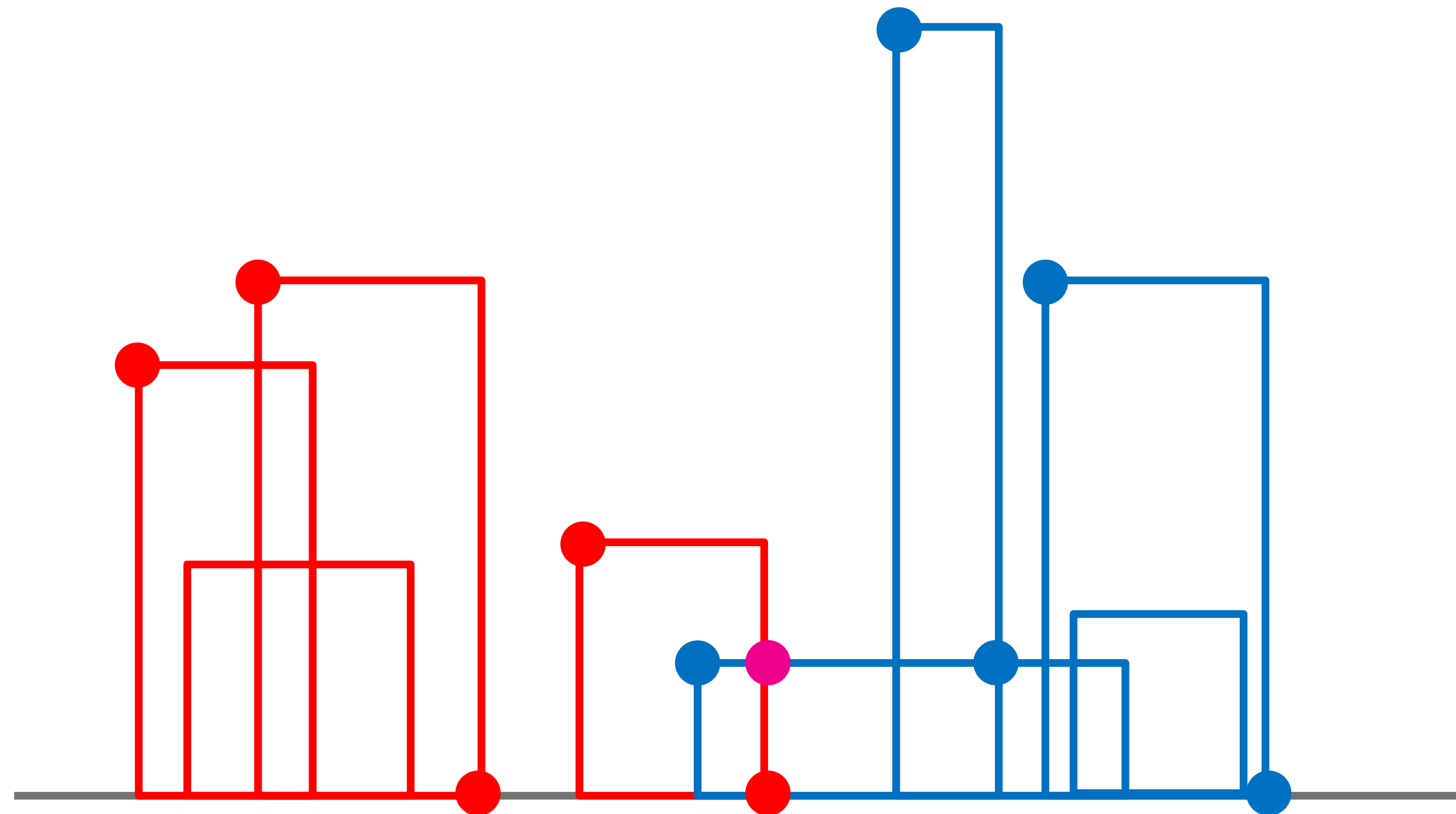


# 스카이라인

15

<https://www.acmicpc.net/problem/1933>

- 높이가 같은데, 연속되는 점을 제거한다.
- 결과 = [(1, 11), (3, 13), (9, 0), (12, 7), (16, 3), (19, 18), (22, 3), (22, 13), (29, 0)]



# 스카이라인

<https://www.acmicpc.net/problem/1933>

- 소스: <http://codeplus.codes/5f1dad235a8d4f85aa826572b8a3dfbb>



# 가장 가까운 두 점

<https://www.acmicpc.net/problem/2261>

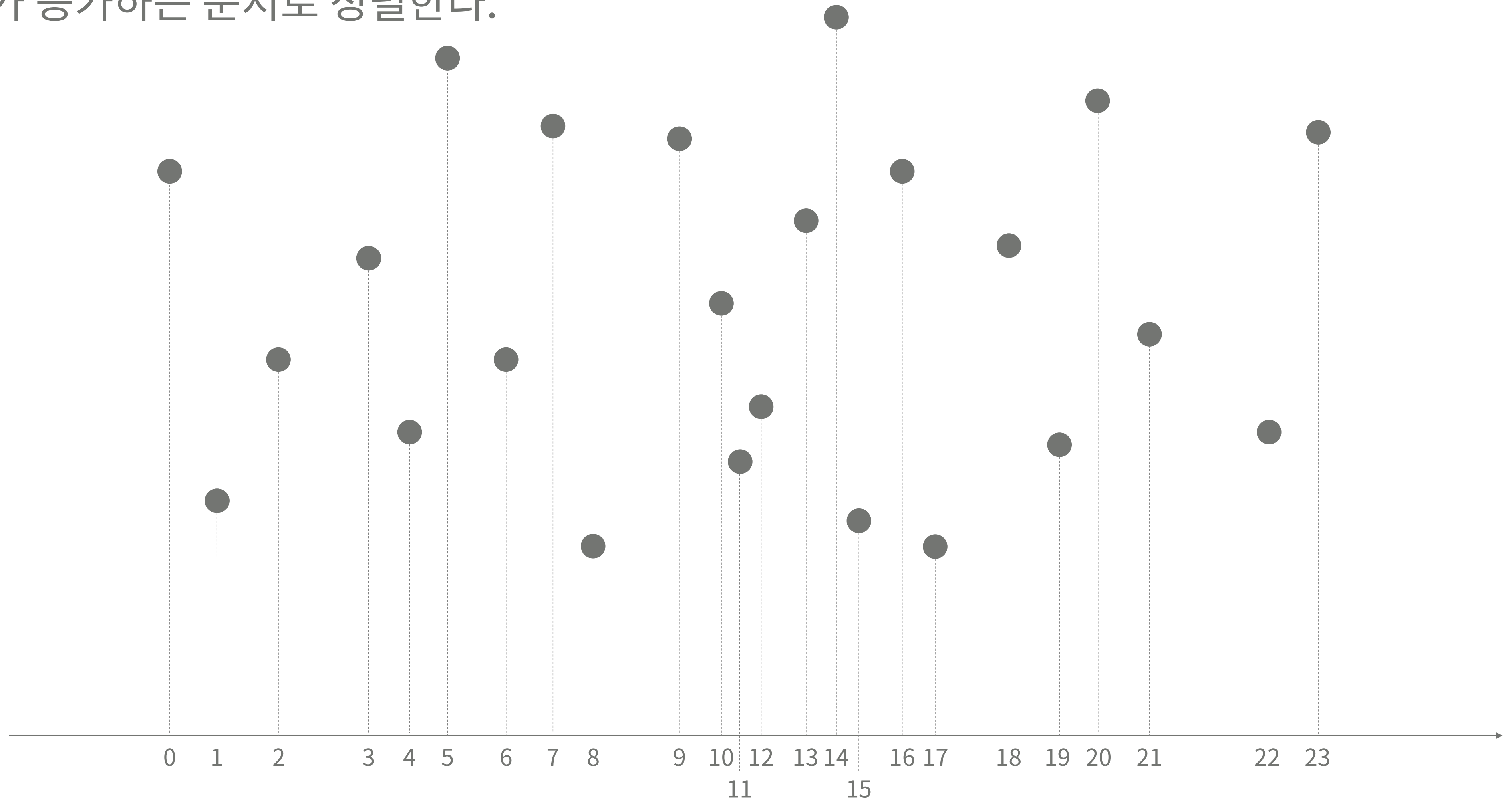
- 2차원 평면 위의 N개의 점 중에서 가장 가까운 두 점을 찾는 문제
- $2 \leq N \leq 100,000$
- 모든 점의 쌍을 조사하면  $O(N^2)$ 이 걸린다.

# 가장 가까운 두 점

18

<https://www.acmicpc.net/problem/2261>

- 먼저 점을 x좌표가 증가하는 순서로 정렬한다.

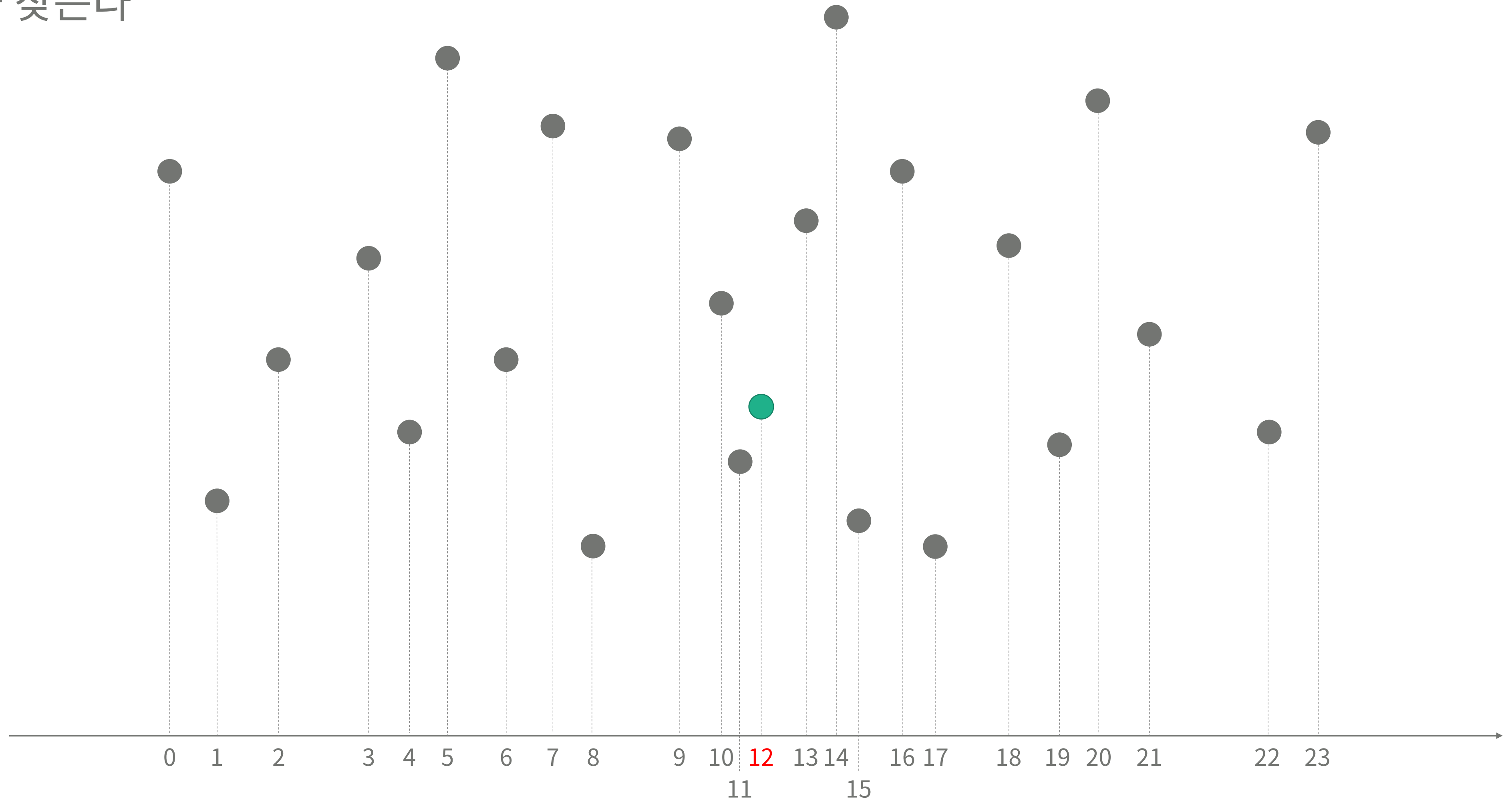


# 가장 가까운 두 점

19

<https://www.acmicpc.net/problem/2261>

- 중간에 있는 점을 찾는다

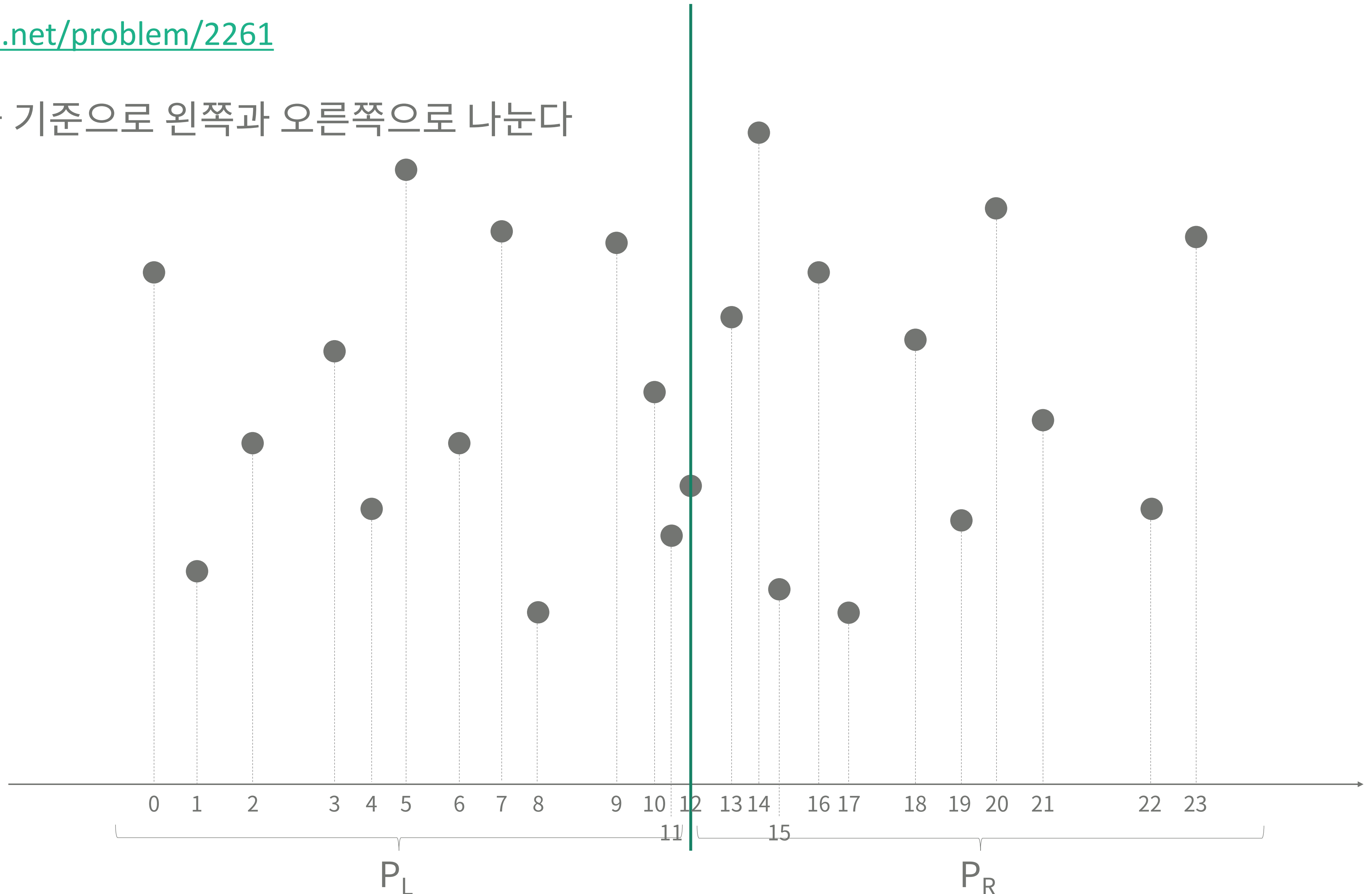


# 가장 가까운 두 점

20

<https://www.acmicpc.net/problem/2261>

- 중간에 있는 점을 기준으로 왼쪽과 오른쪽으로 나눈다
- 왼쪽:  $P_L$
- 오른쪽:  $P_R$



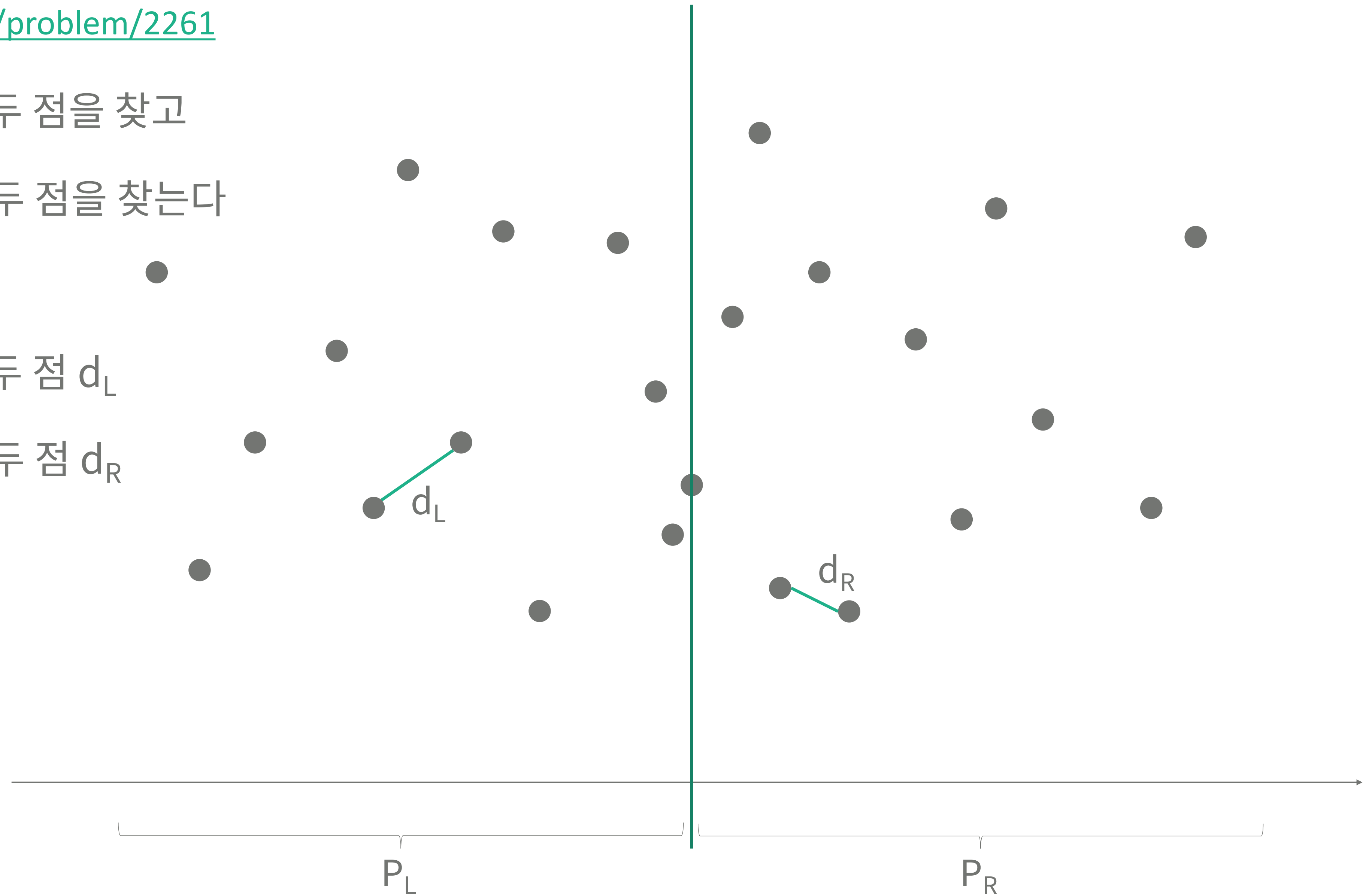
# 가장 가까운 두 점

21

<https://www.acmicpc.net/problem/2261>

- $P_L$ 에서 가장 가까운 두 점을 찾고
- $P_R$ 에서 가장 가까운 두 점을 찾는다

- $P_L$ 에서 가장 가까운 두 점  $d_L$
- $P_R$ 에서 가장 가까운 두 점  $d_R$

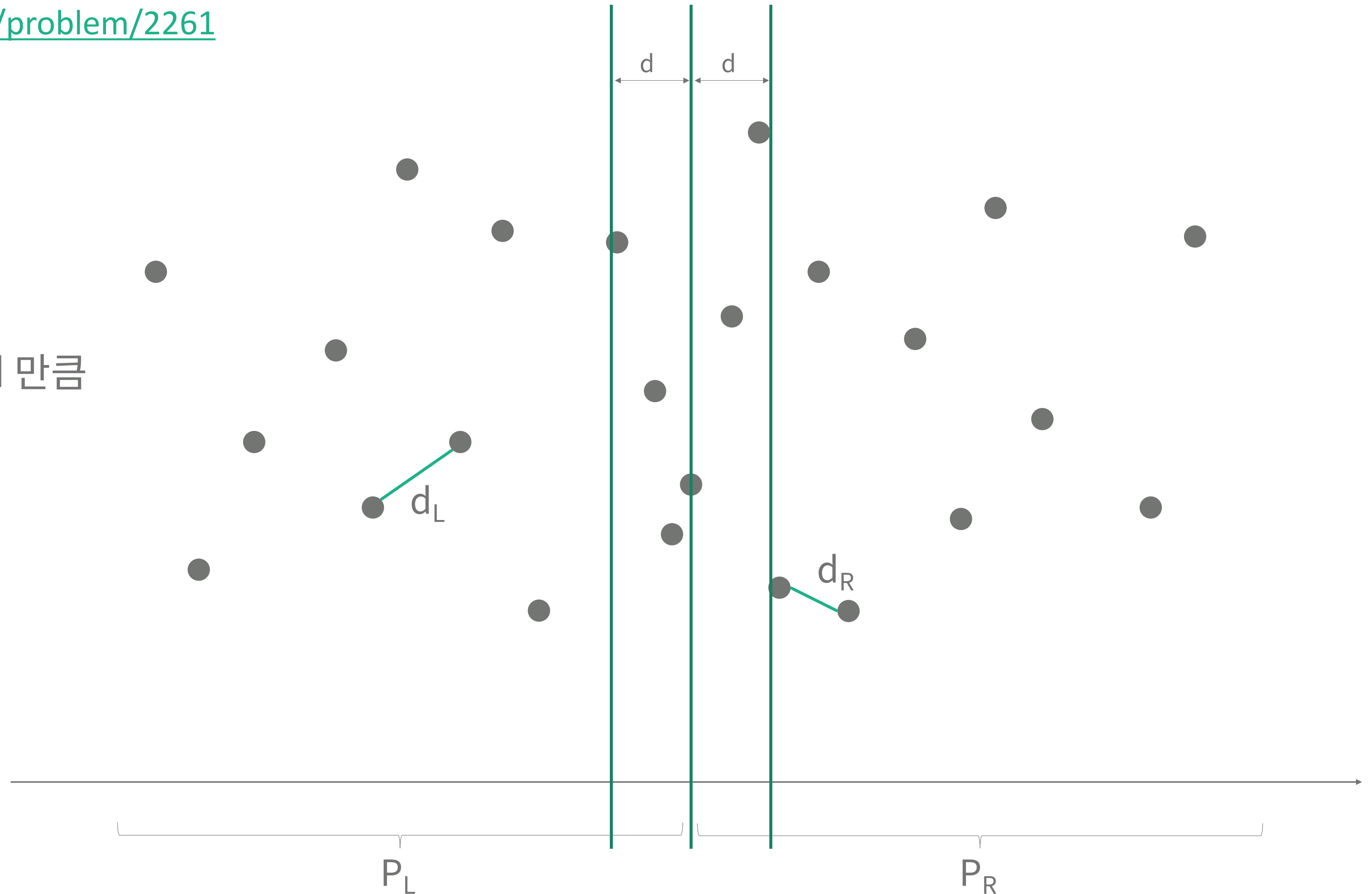


# 가장 가까운 두 점

22

<https://www.acmicpc.net/problem/2261>

- $d = \min(d_L, d_R)$
- 이라고 했을 때
- 가운데 점으로부터
- 가운데로부터  $-d, +d$  만큼
- 떨어진 곳에서
- 가장 가까운 두 점을
- 찾아야 한다

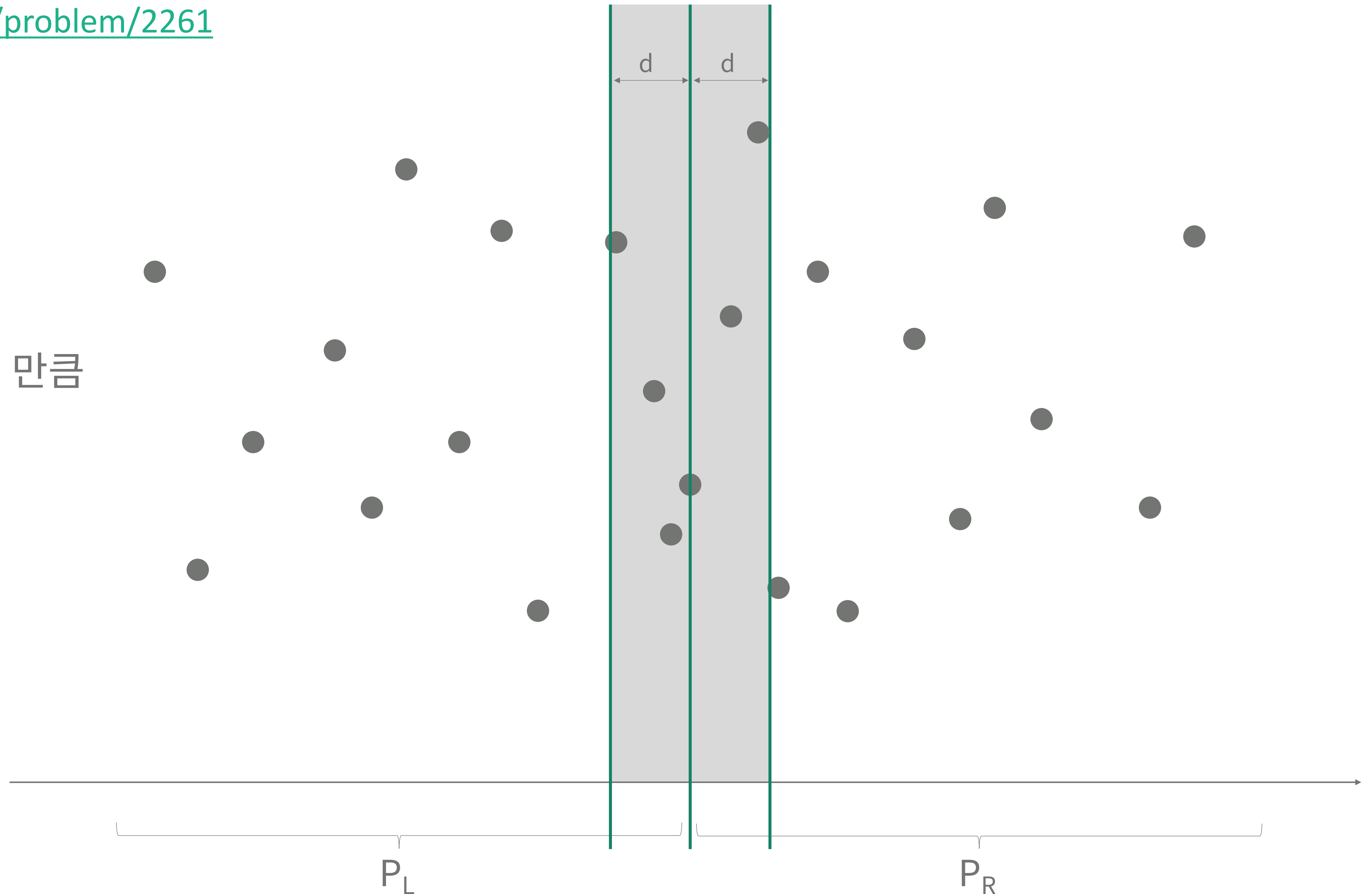


# 가장 가까운 두 점

23

<https://www.acmicpc.net/problem/2261>

- $d = \min(d_L, d_R)$
- 이라고 했을 때
- 가운데 점으로부터
- 가운데로부터  $-d, +d$  만큼
- 떨어진 곳에서
- 가장 가까운 두 점을
- 찾아야 한다

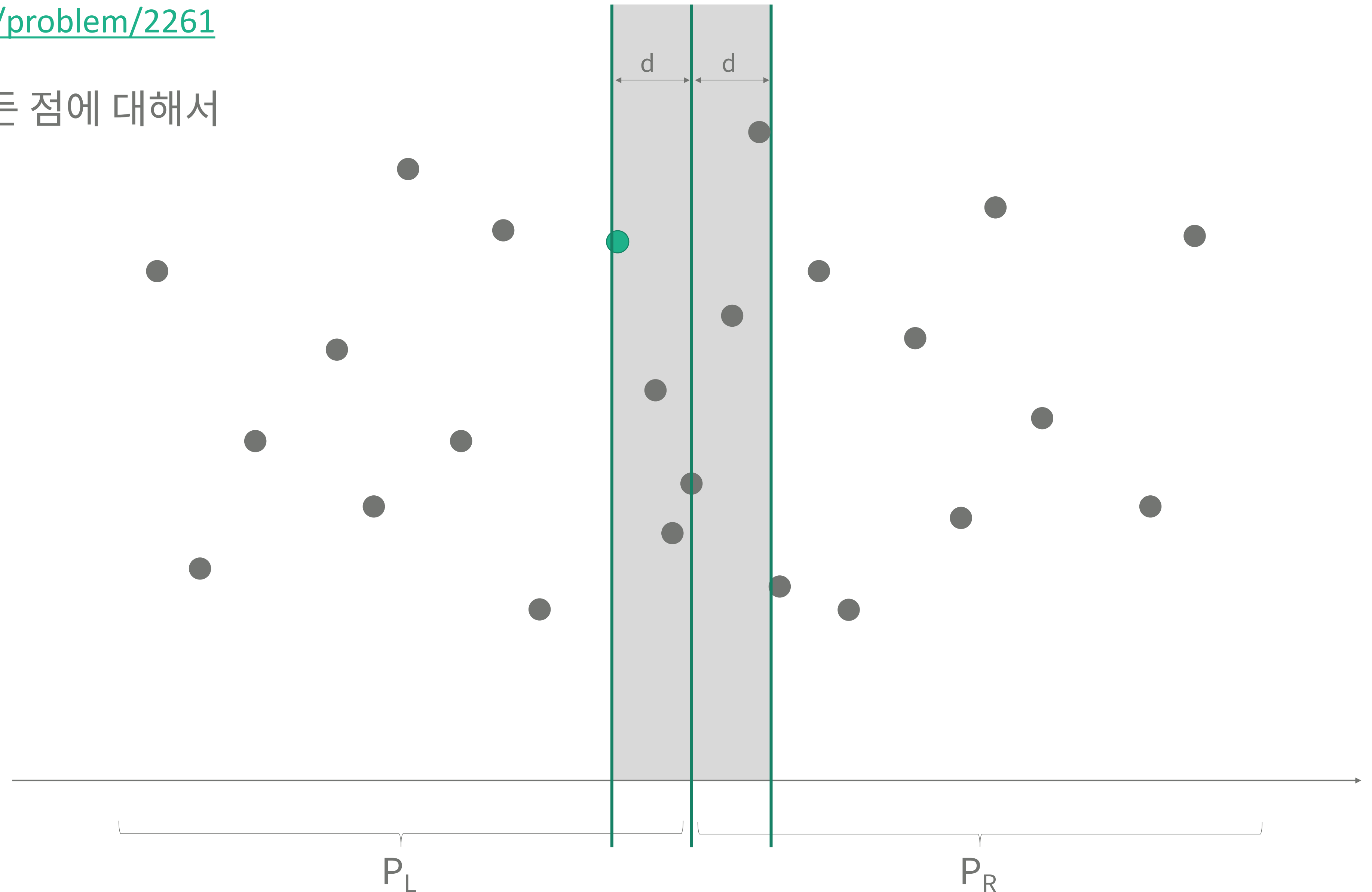


# 가장 가까운 두 점

24

<https://www.acmicpc.net/problem/2261>

- 회색 영역에 있는 모든 점에 대해서
- 가장 가까운 거리를
- 구해야 한다



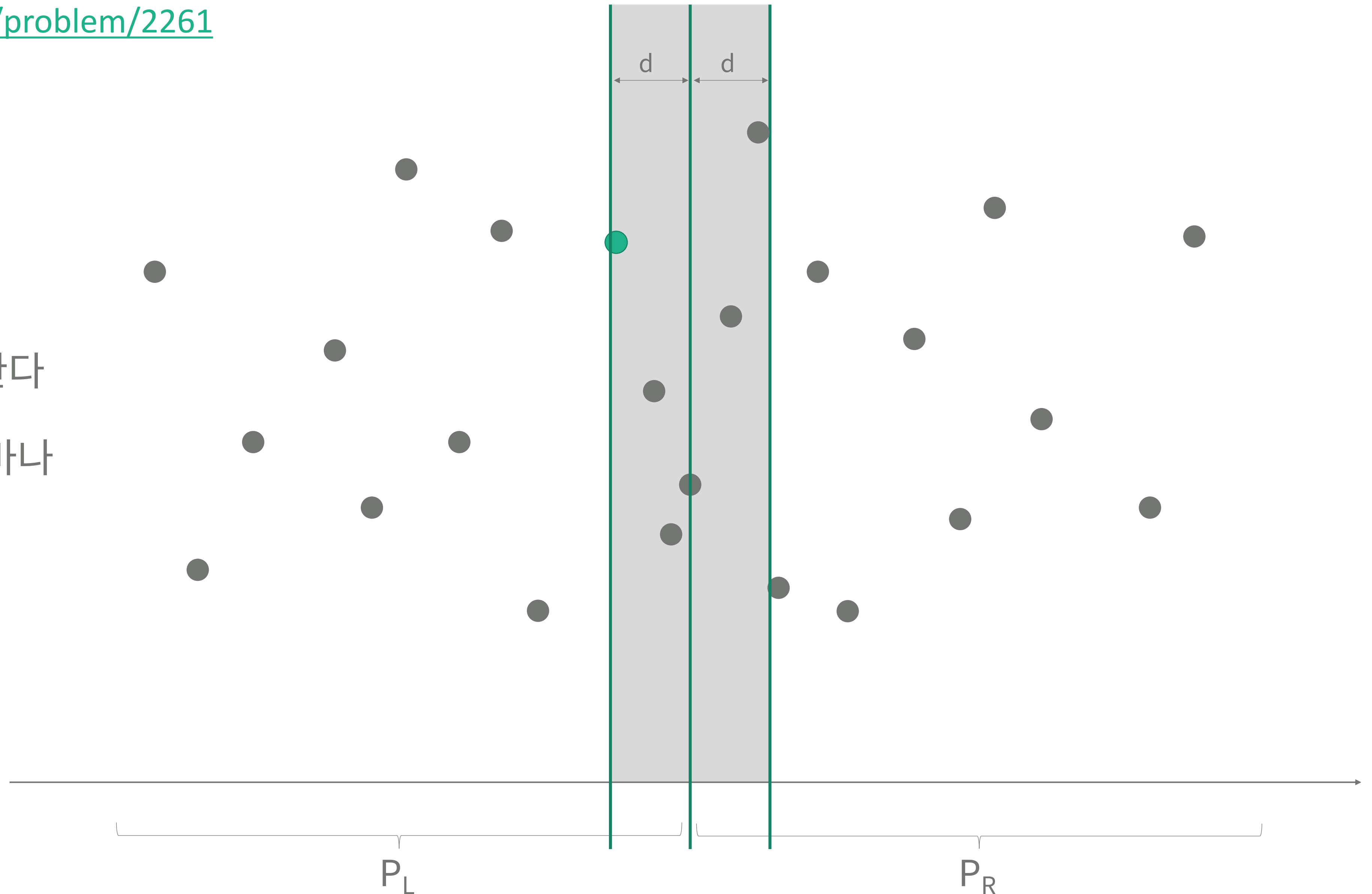


# 가장 가까운 두 점

25

<https://www.acmicpc.net/problem/2261>

- 회색 영역에 있는
- 각각의 점에 대해서
- 다른 모든 점에 대한
- 거리를 조사해봐야 한다
- 이 과정은 시간이 얼마나
- 걸릴까?

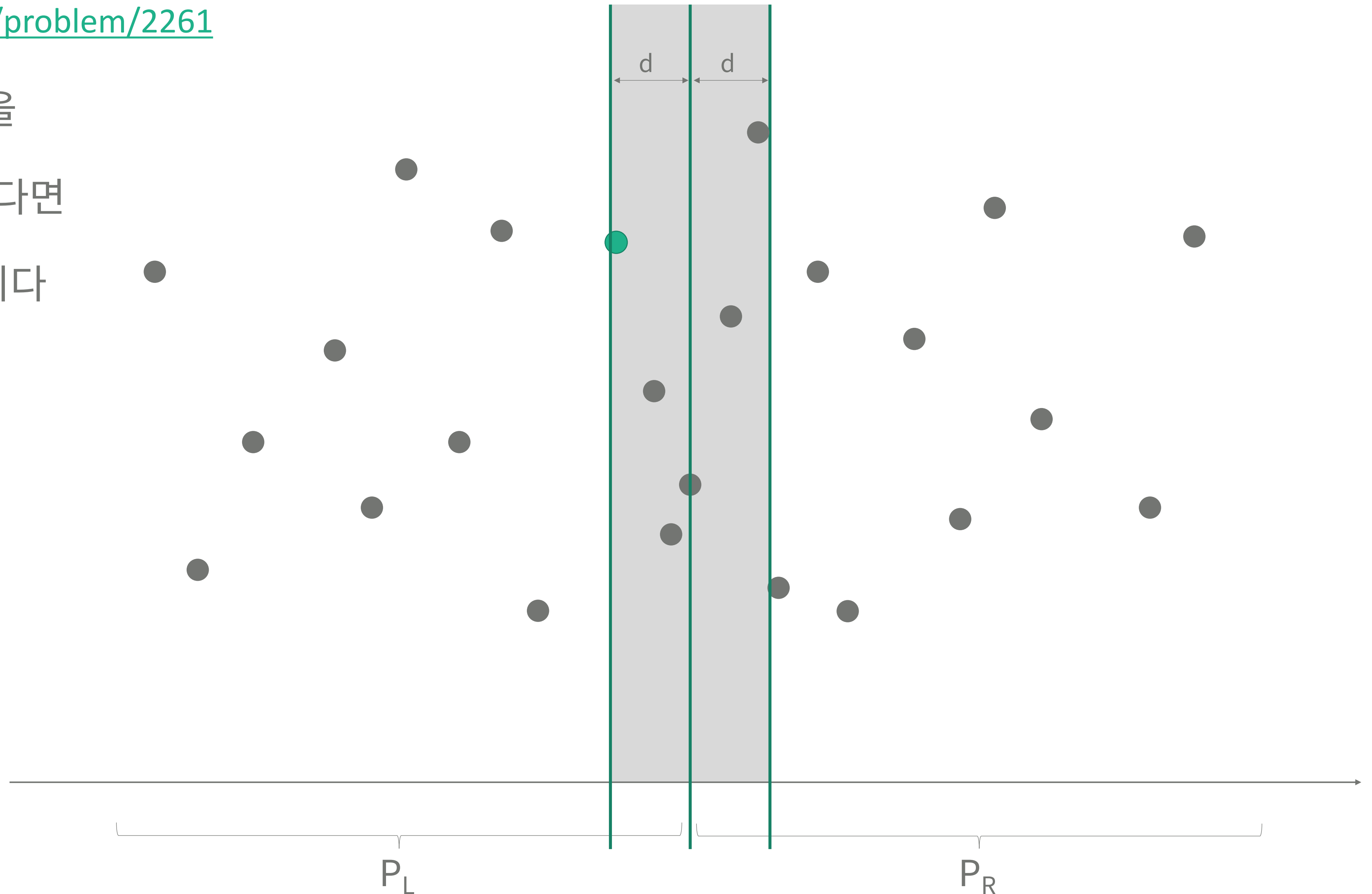


# 가장 가까운 두 점

26

<https://www.acmicpc.net/problem/2261>

- 회색 영역에 있는 점을
- y좌표 순으로 정렬했다면
- $O(N^2)$ 이 아닌  $O(N)$ 이다

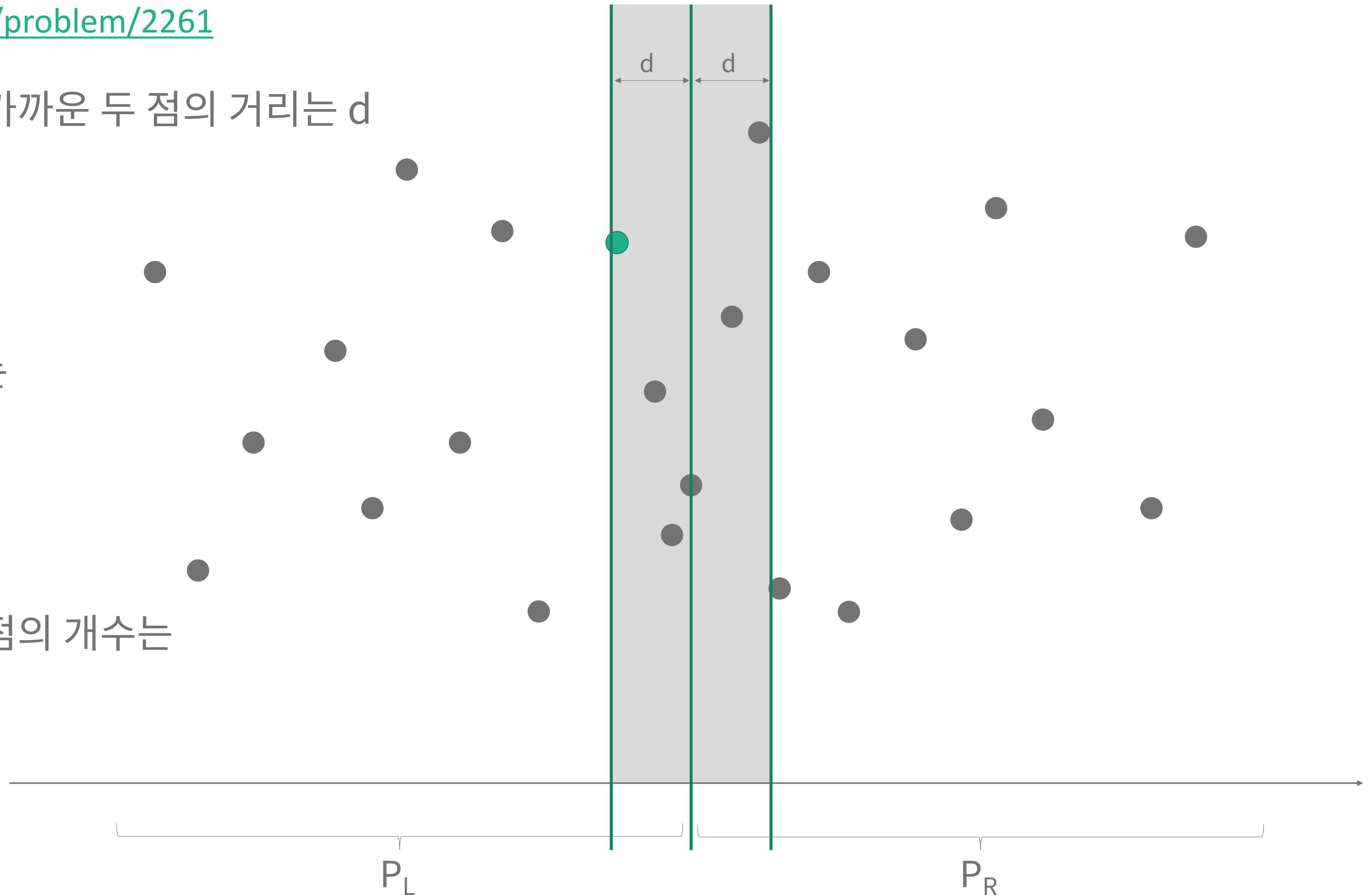


# 가장 가까운 두 점

27

<https://www.acmicpc.net/problem/2261>

- 현재까지 구한 가장 가까운 두 점의 거리는  $d$
- 어떤 점의 좌표가
- $(x, y)$  였다면
- 거리를 조사해야 하는
- 점의  $y$ 좌표 범위는
- $[y-d, y+d]$  이다.
- 이 범위에 들어있는 점의 개수는
- 최대 6개이다.

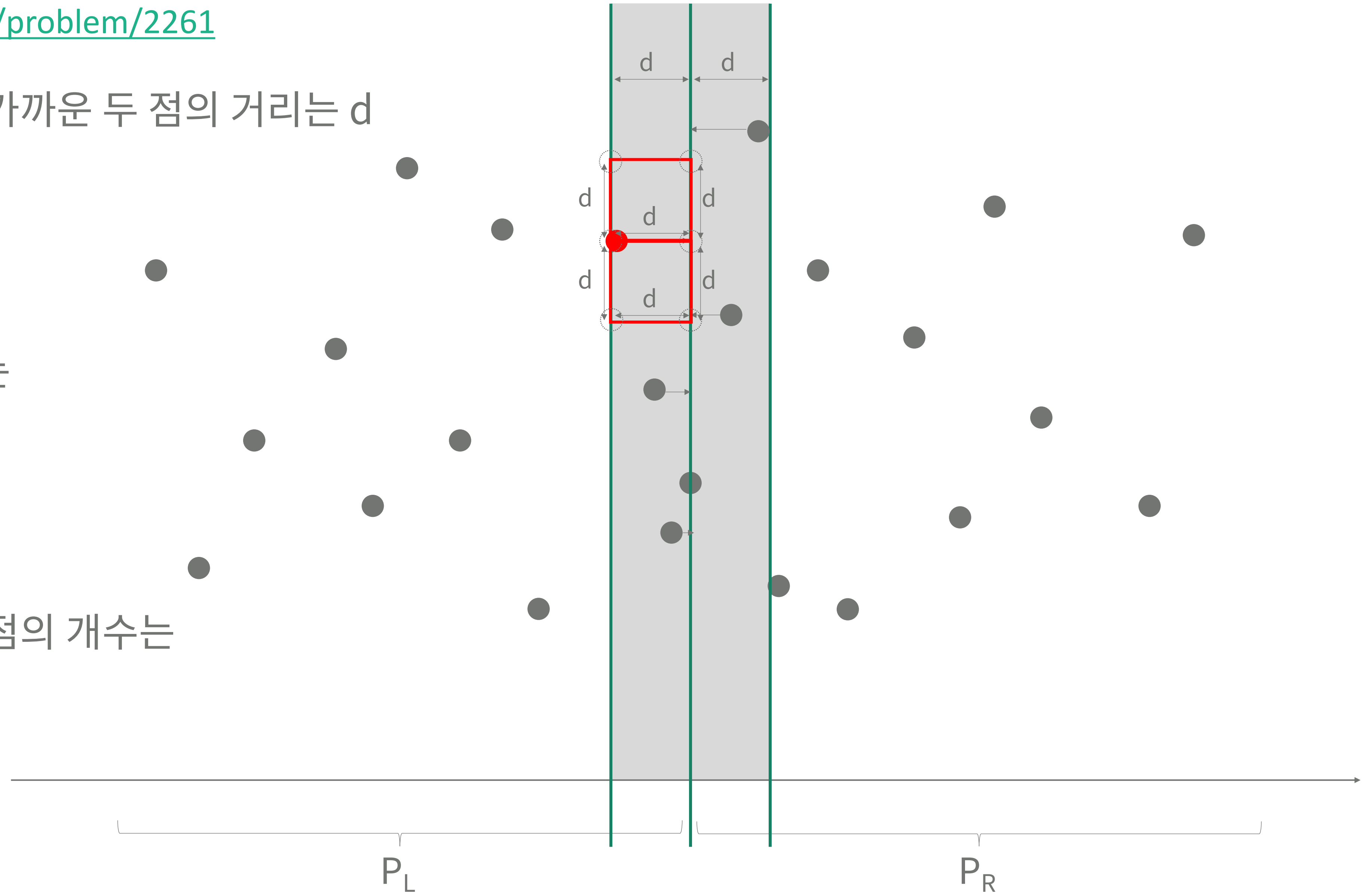


# 가장 가까운 두 점

28

<https://www.acmicpc.net/problem/2261>

- 현재까지 구한 가장 가까운 두 점의 거리는  $d$
- 어떤 점의 좌표가
- $(x, y)$  였다면
- 거리를 조사해야 하는
- 점의  $y$ 좌표 범위는
- $[y-d, y+d]$  이다.
- 이 범위에 들어있는 점의 개수는
- 최대 6개이다.



# 가장 가까운 두 점

29

<https://www.acmicpc.net/problem/2261>

- 시간 복잡도:  $O(N \lg^2 N)$

# 가장 가까운 두 점

30

<https://www.acmicpc.net/problem/2261>

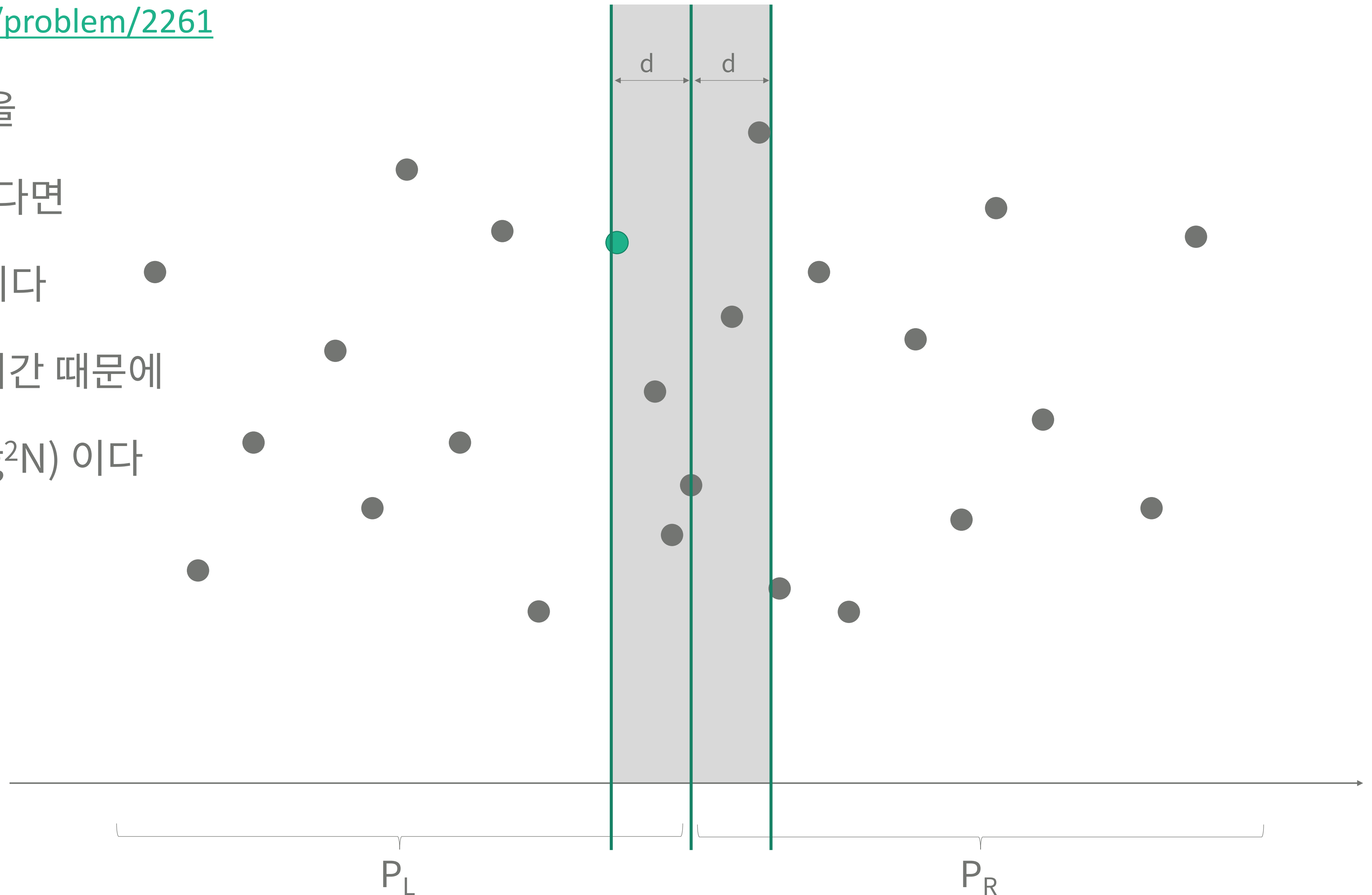
- 소스: <http://codeplus.codes/caab2883bef145de8a2f0c8567c2a193>

# 가장 가까운 두 점

31

<https://www.acmicpc.net/problem/2261>

- 회색 영역에 있는 점을
- **y좌표 순으로 정렬**했다면
- $O(N^2)$ 이 아닌  $O(N)$ 이다
- 정렬하는데 걸리는 시간 때문에
- 시간 복잡도가  $O(N \lg^2 N)$  이다



# 가장 가까운 두 점

<https://www.acmicpc.net/problem/2261>

- 미리 점을  $y$ 순으로 정렬해놓고 상대적인 순서를 유지한채로 나누면 매번 정렬을 하지 않아도 된다.
- 이렇게 구현하면  $O(N \lg N)$ 으로 구현할 수 있다.



# 가장 가까운 두 점

<https://www.acmicpc.net/problem/2261>

- 소스: <http://codeplus.codes/88548eff050141bb8ed1ce938cc5e5c2>