

C9. Interview

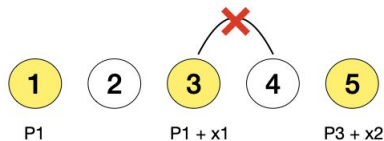
1. 문제 분석

- N 명의 지원자에 대하여 각 지원자 P_i 의 점수가 (x_i, y_i) 로 주어질 때, 다음 조건을 만족하여 만들 수 있는 그룹의 최대 개수를 구하여라
 1. 두 지원자 P_i, P_j 에 대하여, $x_i > x_j$ 와 $y_i < y_j$ 를 만족하거나 $x_i < x_j$ 와 $y_i > y_j$ 를 만족하면 같은 그룹으로 만든다.
 2. 가능한 많은 그룹을 만든다.

2. 문제 해결

- `group[n]` : 지원자가 속한 그룹의 번호
 - `scores[n]` : 지원자의 점수(x_i, y_i)
1. pair 형태로 지원자 N 명의 점수(x_i, y_i)를 입력받아 배열에 저장한다. 초기 그룹 수(groups)는 N 으로 지정한다
 2. x_i 에 대하여 내림차 순으로 정렬한다.
 3. 모든 $P_i, 1 \leq i \leq N-1$ 에 대하여 다음을 반복한다.
 - 3.1. 해당 지원자가 최상위 노드인 경우(`group[i] == i`)
 - 3.1.1. `group[j], i+1 ≤ j ≤ N`와 비교하여 $y_i < y_j$ 라면 두 지원자를 하나의 그룹으로 묶는다. (union-find) 이후 초기 그룹수를 1 감소한다.
 - 3.2. 지원자가 최상위 노드가 아닌 경우(`group[i] != i`)
 - 3.2.1. 3.1.1과정을 건너뛰고 다음 지원자에 대해서 3.1을 만족하는지 확인한다.
 4. 과정3을 마친 뒤 groups를 출력한다.

`group[i] != i` 일때 다른 지원자와의 y비교가 필요없는 이유



Sorting이 되어있는 경우, 만약 3번과 4번이 같은 그룹으로 묶일 수 있다고 가정하면 $P4 > P3$ 이어야하는데, 이 경우 무조건 $P4 > P1$ 이기 때문에 이미 1번 사람을 기준으로 검사할 때 같은 그룹에 속한다. 따라서 3번과 4번이 같은 그룹으로 묶일 수 없으므로 이미 그룹으로 묶인 사람들은 확인할 필요가 없다.

3. 풀이 분석

- Time Complexity
 - 1. input : $O(n)$
 - 2. heap sort : $O(n \log n)$
 - 3. for loop : $O(n^2)$
 - 3.1.1 union-find(optimized) : $O(\log n)$
 best case : groups가 1인 경우 루프 한번에 모든 검사가 끝나므로 $O(n \log n)$ 이다.
 worst case : groups가 n 인 경우 모든 P_i, P_j 에 대해 3.1.1과정을 수행해야 하므로 $O(n^2)$ 이다.
- Space Complexity
 - `groups[n], level[n], scores[n]`: $O(n)$
 - `level[n]` : `group[i]`에 속하는 트리의 깊이

4. 응용 및 discussion

만약 그룹 합병의 조건이 한 그룹의 모든 그룹원 쌍이 조건을 만족해야하는 경우는?