

P10. Dinner Party

이원빈

문제분석 및 아이디어.

입력되는 각 학생 쌍이 서로 다른 방에 배치될 수 있는지를 판단해야하는 문제. 학생 쌍의 학생 각각을 서로 다른 그룹으로 배치하며 전개한다.

각 학생(i)에 대해 해당 학생이 속한 그룹과 속하지 말아야하는 그룹 두 개를 만드는데, 입력되는 두 학생 a, b에 대해 <a가 속한 그룹>은 <b가 속하지 않아야하는 그룹(여집합)>과 같은 그룹이고, b에 대해서도 <b가 속한 그룹>은 <a가 속하지 않아야하는 그룹>과 같은 그룹이어야한다. 그래야 두 학생 a, b가 서로 다른 그룹에 배치된다고 할 수 있다.

만약 입력으로 받은 두 학생이 만약 이미 같은 그룹에 속해 있으면 해당 request의 순서가 곧 거절해야하는 첫 번째 요청이고, 만약 그렇지 않다면 <a가 속한 그룹>과 <b가 속하지 않아야하는 그룹>을 합치고, <b가 속한 그룹>과 <a가 속하지 않아야하는 그룹>을 합친다.

해결방법.

1. 학생 수 n과 학생 쌍의 수 m개를 입력받는다.
2. i번째 학생이 어떤 학생과 같은 그룹을 이루고 있는지를 나타내는 `group[n+1]`과 속하지 않아야하는 그룹 `groupOpposite[n+1]`을 생성한다.
 - a. 예를 들어, `group[1] = 2`, `group[2] = 3`, `group[3] = 3`라면, 1번째 학생은 2번째 학생과 같은 그룹이고, 2번째 학생은 3번째 학생과 같은 그룹이며, 3번째 학생은 스스로와 같은 그룹이므로 세 학생이 모두 같은 그룹이라고 할 수 있다. 또한, `group[3]`이 곧 해당 그룹의 대표값이라고 할 수 있다. 이와 같이 i번째 학생이 속한 그룹의 대표값을 찾는 과정을 **find(i)**라 하자. 처음엔 `group[i] = i`로, `groupOpposite[i] = i + n`으로 초기화한다. (3-a-ii에서의 논리 오류 방지)
3. m개의 학생 쌍(x, y)을 입력 받으며 다음을 반복한다.
 - a. x가 속한 그룹의 대표값과 y가 속한 그룹의 대표값을 찾고, 그 두 값을 비교한다.
 - i. 만약 대표값이 같다면 이미 x와 y가 같은 그룹에 속한 것이므로 해당 쌍이 곧 거절해야할 요청임을 출력하고, 프로그램을 종료한다.
 - ii. 만약 다르다면, x의 그룹과 y가 속하지 않아야하는 그룹을 합치고, y의 그룹과 x가 속하지 않아야하는 그룹을 합친다.
 1. 이는 <x가 속한 그룹>의 '대표값'을 <y가 속하지 않아야하는 그룹>의 '대표값'값으로 설정하는 방식으로 진행한다.
4. 만약 모든 요청이 가능하다면 가능하다는 문구를 출력하고 프로그램을 종료한다.

시간복잡도.

1. $N + 1$ 개의 배열 두 개 생성 및 초기화 -> $O(N)$
2. M개의 각 학생 쌍마다, N명에 대한 find 과정 6번 진행 -> $O(M \log N)$ | 따라서 시간복잡도는 $O(M \log N)$