C7. Medicine

21700140 김유진, 21700477 윤다은

1. Problem Analysis

(All medicines have to taken in two days)

When number of medicines N and M pair of conflicting medicine are given, find the first pair of medicines, which would violate the conflicting list.

- Whenever a pair comes in, divide the medicine into the first and second days, and check the medicine assigned on the same day by using DFS: TLE

→ We used some informations to define the relationship between side effects cases

2. Problem Solving

m_i = medicine number i
root[i] = group number of m_i
day[i] = day to have m_i (1 or 2)

- **Input processing**
  - Store M pair in 2D array form
- **Preprocessing**
  - Initialize root[i] to i ($1 \le i \le N$) and day[1] ~ day[N] to 1
- **Processing for each M pair (m_a, m_b)**

  case1. m_a and m_b are in same group (if root[a] == root[b])
  - i. m_a and m_b are taken in same day (if day[a] == day[b]) ⇒ answer
  - ii. m_a and m_b are taken in not same day(else) ⇒ no side effect (pass)

  case2. m_a and m_b are in different group (else)
  let G_i is group of medicine that has group number i,
  merge($G_{root[a]}$, $G_{root[b]}$) = changing group number of $G_{root[b]}$ to root[a]
  - i. m_a and m_b is taken in same day (if day[a] == day[b])
  ⇒ merge($G_{root[a]}$, $G_{root[b]}$) and change day[i] for $G_{root[b]}$ (1→2, 2→1)
  - ii. m_a and m_b is taken in not same day (else)
  ⇒ merge($G_{root[a]}$, $G_{root[b]}$)

3. Problem Solving Analysis

- **Time complexity: O(MN)**
  - Input processing ⇒ O(M)
  - Preprocessing ⇒ O(N)
  - Processing for each M pair ⇒ O(MN)
    worst case
    : (m_a, m_b) needs merge($G_{root[a]}$, $G_{root[b]}$) for all M pairs
    - case1-i, ii = constant time
    - case2-i, ii = O(MN)

  ∴ O(M) + O(N) + O(MN) = O(MN)

- **Space complexity: O(M+N)**
  - group number of m_i : arr[N]
  - day that have to take m_i: arr[N]
  - M pair: arr[M][2]

  ∴ O(N) + O(N) + O(M) = O(M+N)

4. Discussion

- We can reduce time complexity to O(M×N/2) by creating map for $G_{root[i]}$ (key: root[i], value:$G_{root[i]}$).
- It is useful checking whether team MT can eliminate existing close friends when forming groups.