## **Ví dụ**

NYN

YNY

NYN

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | N | Y | N |
| 1 | Y | N | Y |
| 2 | N | Y | N |

→ 0 là bạn của 1, 1 là bạn của 2 ⇒ 0 và 2 là possible friend.

→ Số possible friend của 0 là 1, số possible friend của 2 cũng là 1.

⇒ Kết quả là người có id nhỏ hơn

⇒ Output: 0 1

NYYY

YNNY

YNNN

YYNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 | N | Y | Y | Y |
| 1 | Y | N | N | Y |
| 2 | Y | N | N | N |
| 3 | Y | Y | N | N |

* 0 không có possible friend nào
* 1:
  + là friend của 0, 0 là friend của 2, 1 và 2 không phải là friend của nhau → 1 và 2 là possible friend

⇒ Đỉnh 1 có 1 possible friend là 2

* 2:
  + là possible friend với 1
  + là friend của 0, 0 là friend của 3, 2 và 3 không phải là friend của nhau → 2 và 3 là possible friend.

⇒ Đỉnh 2 có 2 possible friend là 1 và 3

* 3:
  + là possible friend với 2

⇒ Đỉnh 3 có 1 possible friend là 2

⇒ Đỉnh 2 có nhiều possible friend nhất

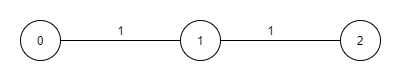
⇒ Output: 2 2

Lưu ý: A là bạn của B, B là bạn của C, C là bạn của D ⇒ A và D là bạn của nhau ⇒ **Không có trường hợp này**

## **Hướng giải:**

* Dùng ma trận kề, với dist[i][j] = 1 khi i và j là bạn của nhau; dist[i][j] = 0 khi i = j; còn lại cho INF.
* Tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh, nếu đường đi này có giá trị là 2 tức là i và j là possible friend.

⇒ Dùng thuật toán Floyd-Warshall.



* Tìm người có số lượng possible friend lớn nhất.

## **Giải thuật:**

B1: Đọc vào ma trận kề matrix

B2: Khởi tạo mảng 2 chiều dist[i][j] = 1 khi i và j là bạn của nhau; = 0 khi i = j và còn lại là INF.

B3: Chạy thuật toán Floyd Warshall để tìm đường đi ngắn nhất giữa mọi cặp đỉnh

B4: Với mọi đỉnh i, đếm số lượng đỉnh j có dist[i][j] = 2. Xuất ra đỉnh có số lượng lớn nhất.

## **Mã giả:**

INF = 1e9

FloydWarshall():

for k = 0 to M - 1:

for i = 0 to M - 1:

for j = 0 to M - 1:

dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])

main():

read(T)

for t = 1 to T:

read(line)

M = len(line)

matrix = []

matrix.append(line)

for i = 1 to M - 1:

read(line)

matrix.append(line)

dist = [][] //mảng 2 chiều M\*M, khởi tạo tất cả phần tử đều là INF

for i = 0 to M - 1:

for j = 0 to M - 1:

if matrix[i][j] == ‘Y’:

dist[i][j] = 1

else if i == j:

dist[i][j] = 0

FloydWarshall() //O(M^3)

maxPossibleFriend = 0

id = 0

for i = 0 to M - 1:

// Đếm số lượng possible friend

count = 0

for j = 0 to M -1:

if dist[i][j] == 2:

count += 1

if count > maxPossibleFriend:

maxPossibleFriend = count

id = i

print(id, maxPossibleFriend)

Độ phức tạp: O(T \* M^3)