

Possible Friends

Big-O Blue - Lecture 11: Floyd-Warshall

Tóm tắt đề bài

Tóm tắt đề bài

Cho một ma trận $M \times M$ với M là số lượng người ($1 \leq M \leq 50$), với (dòng i , cột j) :

- Nếu ô (i, j) có giá trị “Y” \rightarrow i được xem là bạn của j .
- Nếu là “N” thì không là bạn.

Hai người được gọi là “có thể bạn bè” (**possible friends**) nếu họ **không phải bạn bè** và họ **có ít nhất một bạn chung**.

➔ **Yêu cầu:** Tìm người có số lượng **possible friends** nhiều nhất.

Mô tả Input/Output

Input: Dòng đầu chứa số lượng test T

- Mỗi test case chứa một ma trận $M \times M$.
- Dòng i cột j của ma trận chứa một trong hai kí tự 'Y' hay 'N', cho biết người i có phải là bạn của người j không.

Output: Với mỗi test case

- In ra người nào có nhiều **possible friends** nhất với **số lượng possible friends** của người đó.
- Nếu có hai người cùng số lượng thì in ra người nào có chỉ số **ID** nhỏ hơn.

Giải thích ví dụ

Ví dụ

Input

3

NYN

YNY

NYN

NYYY

YNNY

YNNN

YYNN

NNYNNNN

NNYNNNN

YYNYNNN

NNYNYNN

NNNYNYY

NNNNYNN

NNNNYNN

Output

0 1

2 2

3 4

Subtest 1

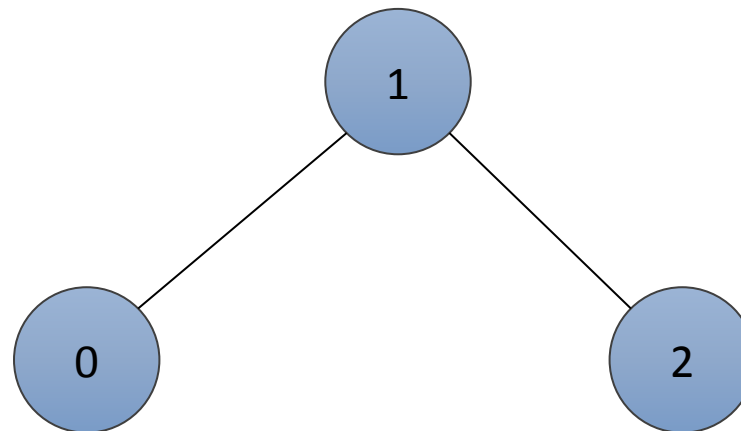
Input

NYN

YNY

NYN

	0	1	2
0		Y	
1	Y		Y
2		Y	



Người 0: Có **1** possible friend là {2}

Người 1: Có 0 possible friend

Người 2: Có **1** possible friend là {0}

Vậy cả người 0 và người 2 đều có số gợi ý kết bạn nhiều nhất nên ta sẽ chọn ưu tiên theo chỉ số i nhỏ nhất

Output: 0 1

Subtest 2

Input

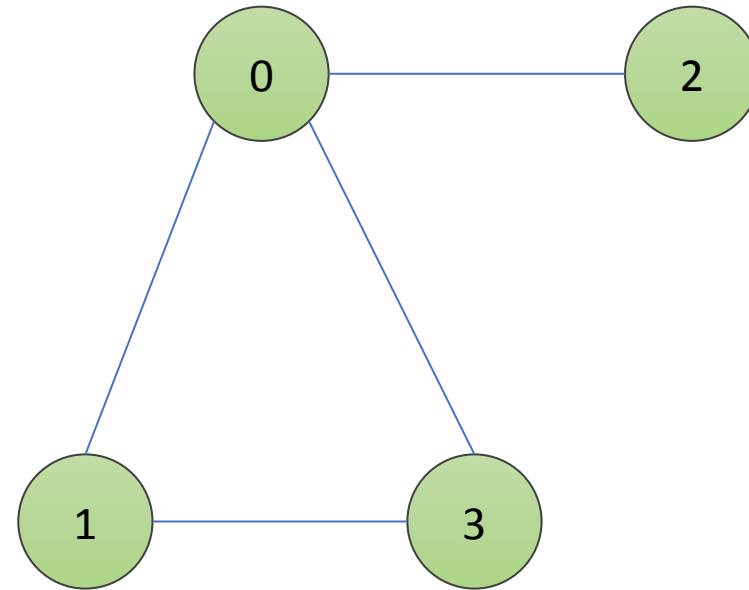
NYYY

YNNY

YNNN

YYNN

	0	1	2	3
0		Y	Y	Y
1	Y			Y
2	Y			
3	Y	Y		



Người 0: Có 0 possible friend

Người 1: Có 1 possible friend {2}

Người 2: Có **2** possible friend {1, 3}

Người 3: Có 1 possible friend {2}

Vậy người 2 có số gợi ý kết bạn nhiều nhất là 2

Output: 2 2

Subtest 3

Input

NNYNNNN

NNYNNNN

YYNYNNN

NNYNYNN

NNNYNYY

NNNNYNN

NNNNYNN

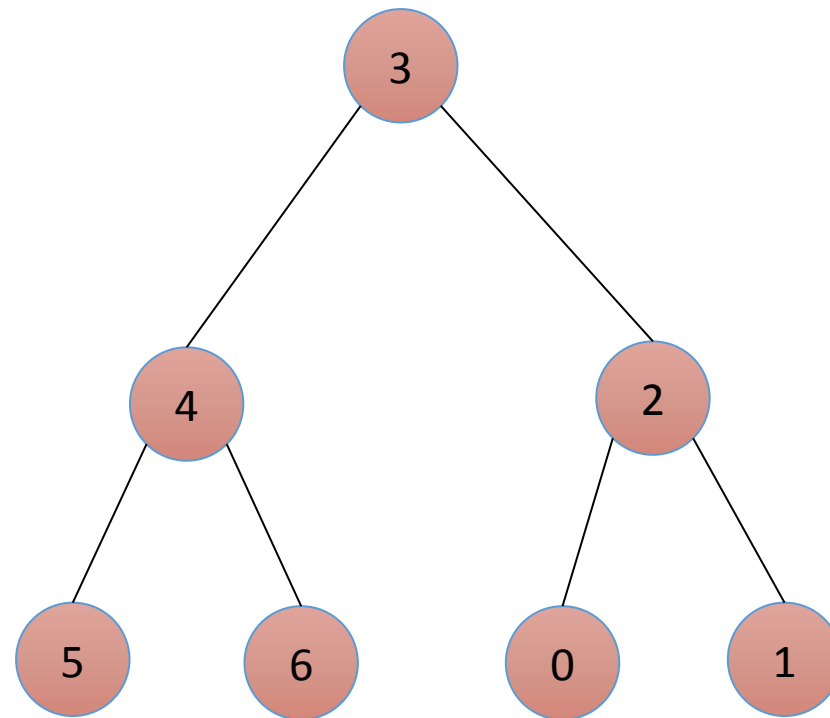
	0	1	2	3	4	5	6
0			Y				
1			Y				
2	Y	Y		Y			
3			Y		Y		
4				Y		Y	Y
5					Y		
6					Y		

Người 0: Có 2 possible friend {1, 3}

Người 1: Có 2 possible friend {0, 3}

Người 2: Có 1 possible friend {4}

Người 3: Có 4 possible friend {0, 1, 5, 6}



Người 4: Có 1 possible friend {2}

Người 5: Có 2 possible friend {3, 6}

Người 6: Có 2 possible friend {3, 5}

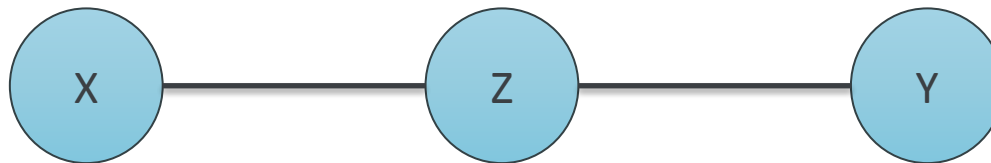
Output: 3 4

Hướng dẫn giải

Nhận xét:

Nếu xem mỗi người là 1 đỉnh của đồ thị thì 2 người X, Y bất kì sẽ là possible friend của nhau khi **khoảng cách** giữa X và Y **đúng bằng 2**.

□ Tìm khoảng cách giữa mọi cặp đỉnh trong đồ thị, đếm possible friend của từng đỉnh và lấy max



Khởi tạo dist

Input

NNYNNNN

NNYNNNN

YYNYNNN

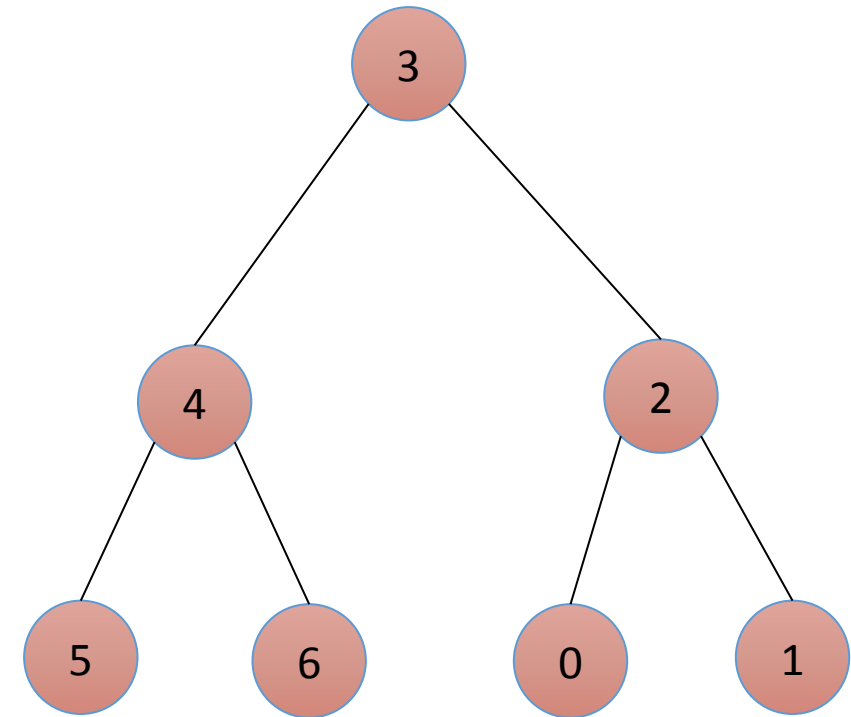
NNYNYNN

NNNYNYY

NNNNYNN

NNNNYNN

	0	1	2	3	4	5	6
0	0	inf	1	inf	inf	inf	inf
1	inf	0	1	inf	inf	inf	inf
2	1	1	0	1	inf	inf	inf
3	inf	inf	1	0	1	inf	inf
4	inf	inf	inf	1	0	1	1
5	inf	inf	inf	inf	1	0	inf
6	inf	inf	inf	inf	1	inf	0



Kết quả dist

Input

NNYNNNN

NNYNNNN

YYNYNNN

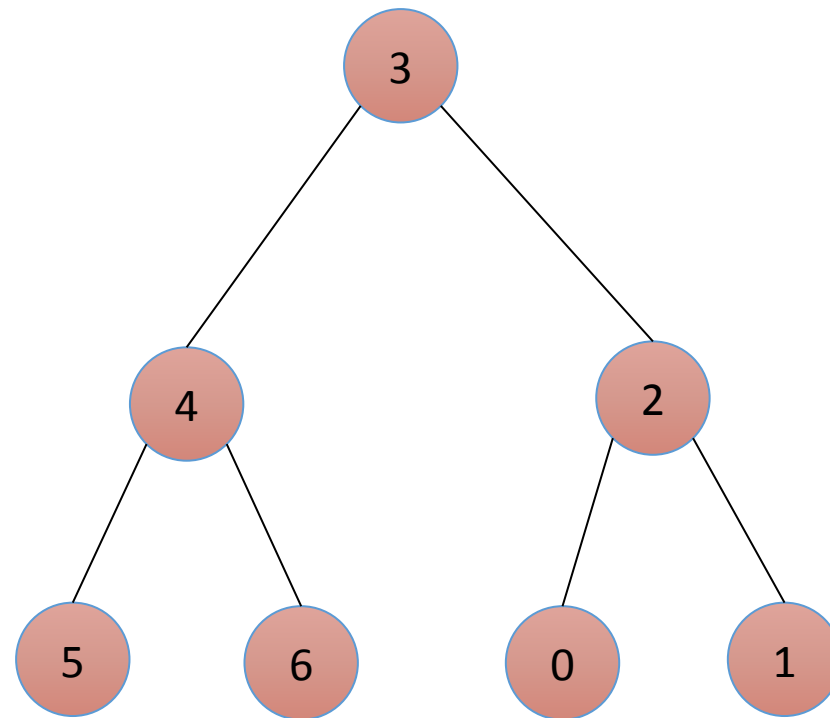
NNYNYNN

NNNYNYY

NNNNYNN

NNNNYNN

	0	1	2	3	4	5	6
0	0	2	1	2	3	4	4
1	2	0	1	2	3	4	4
2	1	1	0	1	2	3	3
3	2	2	1	0	1	2	2
4	3	3	2	1	0	1	1
5	4	4	3	2	1	0	2
6	4	4	3	2	1	2	0



Output: 3 4

Hướng dẫn giải (đối với từng sub test)

B1: Đọc vào ma trận, khởi tạo mảng dist để lưu khoảng cách giữa mọi cặp đỉnh

B2: Chạy thuật toán **Floyd**

B3: Duyệt từng người i:

- + Với từng người j, nếu $\text{dist}[i, j] == 2$ thì tăng số lượng **possible friends** của người i lên 1

- Chạy xong M người j, có được count: số lượng possible friends của người i.

- + Cập nhật lại chỉ số người i và maximum possible friends

B4: In kết quả là chỉ số và maximum possible friends

Độ phức tạp: $O(T \times M^3)$

Mã giả

Mã giả

```
read T
INF = 1e9
for tc = 1 → T:
    graph = []
    read first_line
    M = length(first_line)
    graph.push(first_line)
    for i = 1 → M - 1:
        read line
        graph.push(line)
    dist = [[INF]*M]*M
    for i = 0 → M-1:
        for j = 0 → M-1:
            if graph[i][j] == 'Y':
                dist[i][j] = 1
            if i == j:
                dist[i][j] = 0
```

```
Floyd(dist)
id = 0
max_friends = 0
for i = 0 → M - 1:
    count = 0
    for j = 0 → M - 1:
        if dist[i][j] == 2:
            count += 1
    if count > max_friends:
        max_friends = count
        id = i
print(id, max_friends)
```


Mã giả

```
function Floyd(dist):  
    for k = 0 → M - 1: #đỉnh trung gian  
        for i = 0 → M - 1: #đỉnh bắt đầu  
            if dist[i][k] == INF: continue  
            for j = 0 → M - 1: #đỉnh đến  
                if dist[k][j] != INF and dist[i][k] + dist[k][j] < dist[i][j]:  
                    dist[i][j] = dist[i][k] + dist[k][j]
```

Thank you