War

### Idea

Tạo n bóng cho n người. (x’ = x + n)

setFriends(x, y) => join(x, y), join(x’, y’)

setEnemies(x, y) => join(x, y’), join(x’, y)

areFriends(x, y) => check(x, y)

areEnemies(x, y) => check(x, y’), check(x’, y)

### Code:

/\*

P[i] = -1 => i là nút đại diện

P[i] = j => j là nút cha của i

Int p[10001];

int getParent(int x) {

return p[i] < 0 ? i : p[i] = getParent(i);

}

// kiem tra 2 nút x, y có chung tập hơp

bool check(int x, int y) {

int px = getParent(x);

int py = getPatent(y);

return px == py;

}

void join(int x, int y) {

int px = getParent(x);

int py = getParent(y);

// chua dung optimize by rank

p[x] = y;

}

int main() {

input(n);

while(input(c, x, y)){

if(c == 0 && x == 0 && y == 0) break;

int xx = x+n;

int yy = y+n;

switch c:

case 1:

if(check(x, yy) || check(xx, y)) {

output(-1);

break;

}

Join(x, y);

Join(xx, yy);

Break;

case 2:

if(check(x, y) || check(xx, yy)) {

output(-1);

break;

}

Join(x, yy);

Join(xx, y);

Break;

case 3:

if(check(x, y) || check(xx, yy)) {

output(1);

} else {

output(0);

}

Break;

case 4:

if(check(xx, y) || check(x, yy)) {

output(1);

} else {

output(0);

}

Break;

}

}

Độ phức tạp O((check+join)\*số thao tác) = O(logN \* số thao tác)