

# Celebrity Problem [GFG](#)

In a group of people, a celebrity is a person who is known by everyone but knows no one. Given a matrix representing the relationships between people (1 if person  $i$  knows person  $j$ , 0 otherwise), find the celebrity in the group.

## Example

Matrix:

0 1 0 1

0 0 0 0

1 1 0 1

1 1 0 0

## Output

The celebrity in this group is person 1.

## Approach 1: Function to find the celebrity using brute force approach

In this approach, we loop through each person and check if they are a potential celebrity by verifying whether everyone knows them and they don't know anyone.

### Steps:

1. For each person  $i$  in the group:
  - Check if all other people know person  $i$  ( $\text{candidate}[i][j] == 1$ ) or person  $i$  doesn't know them ( $\text{candidate}[j][i] == 0$ ).
  - If the above condition is met for all other people, return  $i$  as the celebrity index.
2. If no celebrity is found, return -1.

### Time Complexity:

- The approach involves nested loops, where the outer loop iterates through each person and the inner loop iterates through all people.
- **Overall, the time complexity is  $O(n^2)$ , where  $n$  is the number of people in the group.**

### Space Complexity:

- The approach uses only a few variables to store indices.
- **Space Complexity:  $O(1)$ .**

### **Approach 2: Function to find the celebrity using stack approach**

In this approach, we use a stack to process pairs of people. We start with all people in the stack. For each pair, we compare whether one person knows the other. If person1 knows person2, we push person2 onto the stack; otherwise, we push person1 onto the stack. Eventually, only the potential celebrity remains in the stack.

#### **Steps:**

1. Push all people onto the stack.
2. Continuously compare pairs of people:
  - Get person1 and person2 from the stack.
  - If person1 knows person2, push person2 onto the stack; otherwise, push person1 onto the stack.
3. The remaining person in the stack is the potential celebrity.
4. Check if the potential celebrity knows others or is known by others. If they do, return -1.
5. Return the index of the potential celebrity.

#### **Time Complexity:**

- The approach involves pushing all people onto the stack and continuously comparing pairs of people.
- **Time Complexity:  $O(n)$ , where  $n$  is the number of people in the group.**

#### **Space Complexity:**

- The approach uses additional space for the stack and a few variables.
- **Space Complexity:  $O(n)$ , where  $n$  is the number of people in the group.**

### **Approach 3: Function to find the celebrity using two-pointer approach**

In this approach, we use a two-pointer technique to narrow down the potential celebrity by comparing their relationships with others. We start with left pointing to the first person and right pointing to the last person. If left knows right, we move left to the right; otherwise, we move right to the left. Eventually, only the potential celebrity remains.

#### **Steps:**

1. Initialize left and right pointers.
2. Continuously compare relationships between the persons pointed to by left and right:

- If left knows right, move left to the right; otherwise, move right to the left.
3. The remaining person at the pointer is the potential celebrity.
  4. Check if the potential celebrity knows others or is known by others. If they do, return -1.
  5. Return the index of the potential celebrity.

**Time Complexity:**

- The approach involves continuously narrowing down the potential celebrity using the two-pointer technique.
- **Time Complexity:  $O(n)$ , where  $n$  is the number of people in the group.**

**Space Complexity:**

- The approach uses only a few variables to store indices.
- **Space Complexity:  $O(1)$ .**