Colored travel

Submit solution

All submissions
Best submissions

Points: 100 (partial)
Time limit: 1.0s
Memory limit: 256M

Problem type

Allowed languages
C, C++

Problem Statement

There is a simple undirected graph with N vertices numbered 1 through N and M edges numbered 1 through M. Edge i connects vertex u_i and vertex v_i. Every vertex is painted either red or blue. The color of vertex i is represented by C_i; vertex i is painted red if C_i = 0 and blue if C_i = 1.

Initially, Tanay is on vertex 1 and Arush is on vertex N. They may repeat the following move zero or more times:

1. Each of them simultaneously moves to an adjacent vertex.
2. The vertices that Tanay and Arush move to must have different colors.

Determine whether Tanay and Arush can end up with Tanay on vertex N and Arush on vertex 1 simultaneously. If it is possible, compute the minimum number of moves required; otherwise, output -1.

You are given T test cases; solve each independently.

Input

- The first line contains an integer T (1 <= T <= 1000), the number of test cases.
- The sum of all N over all test cases does not exceed 2000.
- The sum of all M over all test cases does not exceed 2000.

Each test case consists of:

Copy

```
C_1 C_2 ... C_N
u_1 v_1
u_2 v_2
...
u_M v_M
```

- 2 <= N <= 2000
- 1 <= M <= min((N*(N-1))/2, 2000)
- C_i = 0 or 1
- 1 <= u_i, v_i <= N
- The graph is simple (no self-loops or multiple edges).
- For the first Batch (30 Pts) the graph has bipartite colouring.

Output

For each test case, output a single integer on its own line:

- The minimum number of moves required for Tanay to reach N and Arush to reach 1 simultaneously, or
- -1 if such a sequence of moves does not exist.

Sample Input

Copy

```
4 4
0 1 0 1
1 2
2 3
1 3
2 4

3 3
0 1 0
1 2
2 3
1 3

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

Sample Output

Copy

```
3
-1
3
```

Explanation of Sample 1

For the first test case, they can swap positions in 3 moves:

1. Tanay: 1 → 3 (blue), Arush: 4 → 2 (red)
2. Tanay: 3 → 2 (red), Arush: 2 → 3 (blue)
3. Tanay: 2 → 4 (blue), Arush: 3 → 1 (red)

Note that in move 1 it is invalid for both to move to the same vertex.

Clarifications

Report an issue

No clarifications have been made at this time.

Hello, **2024101067**.