



มัลติเวิร์ส (multiverse)

Problem by : JO

Problem : จงหาจำนวนสีที่น้อยที่สุดในการทำให้โลกที่ติดกันมีสีที่ต่างกัน

Tag : Graph, Greedy

เราสามารถมองโลกแต่ละโลกให้เป็น node ซึ่งมี directed edge ไปหาโลกที่ a_i และเนื่องจาก a_i จะไม่ซ้ำกันเลย แสดงว่าทุก node จะมี 1 indegree และ 1 outdegree ซึ่งเราสามารถพิสูจน์ได้ว่า node ทุก node จะอยู่ใน cycle และจะอยู่เพียง cycle เดียวเท่านั้น

ต่อไปเราจะสนใจทีละ cycle โดยทำการเลือกสีให้แต่ละ node ที่ติดกันใน cycle ให้มีสีที่ต่างกันแบบ Greedy กล่าวคือเลือกสีสลับเบอร์ 1, 2, 1, 2 ไปเรื่อยๆ จนกว่าจะกลับมาที่จุดเริ่มต้น โดยเราสามารถ observe ได้ว่าสีที่ต่างกันมากที่สุดจะมีเพียง 3 สำหรับแต่ละ cycle จะมีเพียง 3 สี

1. ใช้ 1 สีเมื่อขนาดของ cycle เป็น 1
2. ใช้ 2 สีเมื่อขนาดของ cycle เป็นเลขคู่
3. ใช้ 3 สีเมื่อขนาดของ cycle เป็นเลขคี่ที่ไม่ใช่ 1

สุดท้ายนี้คำตอบของเราคือค่า max ของจำนวนสีที่ใช้สำหรับทุกๆ cycle

หากข้อมูลนำเข้าเป็นดังต่อไปนี้

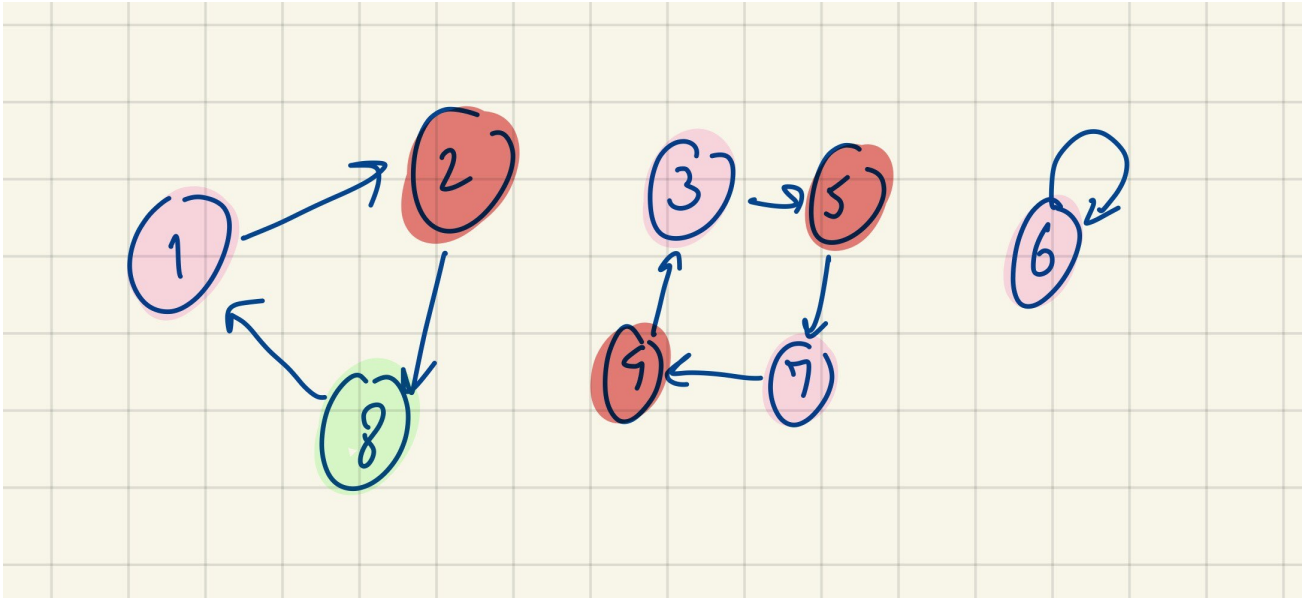
```
1
8
2 8 5 3 7 6 4 1
```

$Q = 1, N = 5, a = [2, 8, 5, 3, 7, 6, 4, 1]$

คำตอบที่ถูกต้องคือ

```
3
```

เราจะได้กราฟที่ลงสีแล้วดังรูปด้านล่าง



ซึ่งจำนวนสีที่แตกต่างกันทั้งหมดเท่ากับ 3 จึงตอบ 3

Solution Code

```
#include <bits/stdc++.h>
using namespace std;

void solve() {
    int n; cin >> n;

    vector go(n+5);
    for (int i = 1; i <= n; i++) cin >> go[i];

    vector vi(n+5);
    int ans = -1;
    for (int i = 1; i <= n; i++) {
        if (vi[i]) continue;

        int u = i;
        int sz = 0;
        do {
            vi[u] = true;
            u = go[u];
            sz++;
        } while (u != i);

        int res = -1;
        if (sz == 1) res = 1;
```

```

        else if (sz%2 == 0) res = 2;
        else res = 3;

        ans = max(ans, res);
    }

    cout << ans << "\n";
}

int main(void) {
    ios_base::sync_with_stdio(0); cin.tie(0);

    int Q; cin >> Q;
    while (Q--) {
        solve();
    }

    return 0;
}

```

Time Complexity : $\mathcal{O}(QN)$