



Национальный исследовательский университет ИТМО
(Университет ИТМО)

Факультет систем управления и робототехники

Дисциплина: Алгоритмы и структуры данных
Отчет по практической работе (1080 задача).

Студент:
Евстигнеев Дмитрий
Группа: *R3242*
Преподаватель:
Тропченко Андрей Александрович

Санкт-Петербург
2021

Цель: написать программу для решения задачи №1080 на [сайте Timus Online](https://timusonlinejudge.ru/)

Задача:

1080. Раскраска карты

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Рассмотрим географическую карту с N странами, занумерованными от 1 до N ($0 < N < 99$). Для каждой страны известны номера соседних стран, т.е. имеющих общую границу с данной. Из каждой страны можно попасть в любую другую, перейдя некоторое количество границ. Напишите программу, которая определит, возможно ли покрасить карту только в два цвета — красный и синий — так, что если две страны имеют общую границу, их цвета различаются. Цвет первой страны — красный. Ваша программа должна вывести одну возможную раскраску для остальных стран или сообщить, что такая раскраска невозможна.

Исходные данные

В первой строке записано число N . Из следующих N строк i -я строка содержит номера стран, с которыми i -я страна имеет границу. Каждое целое число в i -й строке больше, чем i , кроме последнего, которое равно 0 и обозначает конец списка соседей i -й страны. Если строка содержит 0, это значит, что i -я страна не соединена ни с одной страной с большим номером.

Результат

Вывод содержит ровно одну строку. Если раскраска возможна, эта строка должна содержать список нулей и единиц без разделителей между ними. i -я цифра в этой последовательности обозначает цвет i -й страны. 0 соответствует красному цвету, единица — синему. Если раскраска невозможна, выведите целое число -1 .

Пример

исходные данные	результат
3 2 0 3 0 0	010

Принято системой (JUDGE_ID: 231802FR):

Решение на языке C++:

```
#include <stdio.h>
#include <vector>
#include <queue>
using namespace std;

struct node_t {
    bool isCheched;
    bool color;
```

```

        vector<int> neighbors;
};

node_t nodes[99];
queue<int> q;
int n;
static int getNextX();
bool isColoringPossible();

int main() {
    scanf("%d", &n);

    for (int i = 1; i <= n; ++i) {
        int x;
        while ((x = getNextX()) != 0) {
            nodes[i].neighbors.push_back(x);
            nodes[x].neighbors.push_back(i);
        }
    }

    q.push(1);
    nodes[1].isChecked = true;
    nodes[1].color = 0;

    if (n != 0 && isColoringPossible()) {
        for (int i = 1; i <= n; ++i) {
            printf("%d", nodes[i].color);
        }
    } else {
        printf("-1");
    }

    getchar();
    getchar();
}

bool isColoringPossible() {
    while (!q.empty()) {
        int curr = q.front();
        q.pop();

        for (int i = 0; i < nodes[curr].neighbors.size(); ++i) {
            int next = nodes[curr].neighbors[i];
            if (!nodes[next].isChecked) {
                nodes[next].isChecked = true;
                nodes[next].color = nodes[curr].color == 0 ? 1 : 0;
                q.push(next);
            }
            else if (nodes[next].color == nodes[curr].color) {
                return false;
            }
        }
    }

    return true;
}

static int getNextX() {
    int x;
    scanf("%d", &x);
    return x;
}

```

Суть алгоритма:

Проверка графа на двудольность.

Считываем количество стран, а также граничащие страны с каждой страной. Первую страну красим в синий цвет. Далее для каждой страны: при переходе в новую страну мы пытаемся ее покрасить в противоположный от предыдущей страны цвет. Если мы переходим в страну, которую уже покрасили, то проверяем, что мы перешли в другой цвет. Если не перешли, то покрасить невозможно и мы выводим -1

Если дошли до конца и все хорошо, то выводим последовательность цветов для раскраски.

Примеры работы программы:

```
3
2 0
3 0
0
010
```