

**МИНОБРНАУКИ РОССИИ**  
**ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ПРИБЛИЖЕННЫЙ АЛГОРИТМ ДЛЯ ЗАДАЧИ О САМОМ ДЛИННОМ  
ПРОСТОМ ЦИКЛЕ**

**ЛАБОРАТОРНАЯ РАБОТА**

студента 3 курса 331 группы  
специальности 100501 — Компьютерная безопасность  
факультета КНиИТ  
Окунькова Сергея Викторовича

Проверил  
доцент

\_\_\_\_\_

А. Н. Гамова

## **СОДЕРЖАНИЕ**

1	Описание алгоритма .....	3
2	Эффективность алгоритма.....	4
3	Реализация .....	5
4	Тестирование программы .....	7
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	8

## **1 Описание алгоритма**

На вход подается граф. Необходимо найти в заданном графе просто цикл максимальной длины.

Данная задача является NP полной, поэтому она не решается за полиномиальное время. Однако, ее можно свести к более простой задаче, ограничив ее тем, что на вход подается невзвешенный граф. Тогда ее можно решить с помощью модифицированного обхода в глубину за полиномиальное время. Достаточно просто найти все простые циклы на графе и вернуть цикл, который проходит через самое большое количество вершин.

## **2 Эффективность алгоритма**

Сложность работы алгоритма определяется как  $O(n + m)$ , так как он основан на обходе графа в глубину.

### 3 Реализация

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

vector<int> path;
vector<vector<int>> ans;

void graph_init(vector<vector<int>> &graph) {
    for (int i = 0; i < graph.size(); ++i) {
        for (int j = 0; j < graph[i].size(); ++j) {
            cin >> graph[i][j];
        }
    }
}

void dfs(vector<vector<int>> &graph, vector<bool> &used, int v, int start) {
    if(used[v]) {
        if(v == start) {
            vector<int> path1 = path;
            path1.push_back(v);
            ans.push_back(path1);
        }
        return;
    }

    used[v] = 1;

    path.push_back(v);
    for(int i = 0; i < graph[v].size(); ++i)
        if(graph[v][i] == 1)
            dfs(graph, used, i, start);

    used[v] = 0;

    path.pop_back();
}

int main() {
```

```

int n;
cin >> n;
vector<vector<int>> graph(n, vector<int> (n));
graph_init(graph);
vector<bool> used(n, false);

for(int i = 0; i < n; i++) {
    used.resize(n, 0);
    dfs(graph, used, i, i);
    path.clear();
}

sort(ans.begin(),ans.end(),
    [](vector<int>& a, vector<int>& b)
    { return a.size() > b.size(); });
cout << "The longest cycle:\n";
for (auto elem : ans[0]) {
    cout << elem << ' ';
}
}

```

#### 4 Тестирование программы

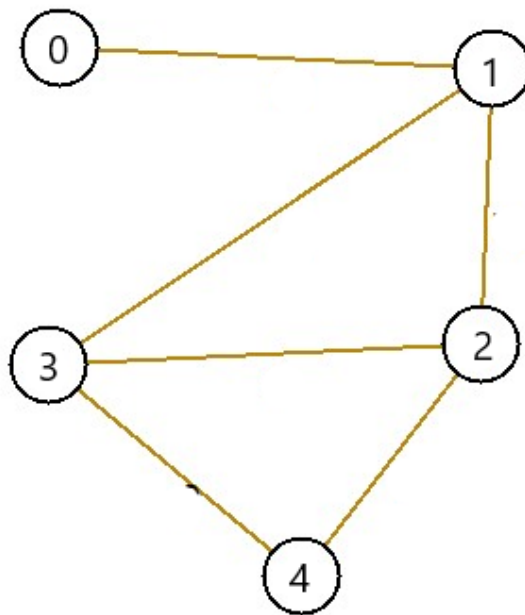


Рисунок 1 – Граф для теста 1

```
5
0 1 0 0 0
1 0 1 1 0
0 1 0 1 1
0 1 1 0 1
0 0 1 1 0
The longest cycle:
3 4 2 1 3
```

Рисунок 2 – Тест 1

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1 Скиена Стивен "Алгоритмы. Руководство по разработке 2018 год. Яз. рус.
- 2 Нииколаус Вирт "Алгоритмы и структуры данных 2008 год. Яз. рус.