

34. Топологическая сортировка

| | |
|---------------------|----------------------------------|
| Ограничение времени | 2 секунды |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Дан ориентированный граф. Необходимо построить топологическую сортировку.

Формат ввода

В первой строке входного файла два натуральных числа N и M ($1 \leq N, M \leq 100\,000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат вывода

Выведите любую топологическую сортировку графа в виде последовательности номеров вершин (перестановка чисел от 1 до N). Если топологическую сортировку графа построить невозможно, выведите -1.

Пример

| | |
|---|-------------|
| Ввод | Вывод |
| 6 6 1 2 3 2 4 2 2 5 6 5 4 6 | 4 6 3 1 2 5 |

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <iterator>
3 #include <vector>
4
5 bool bad_graph = false;
6 std::vector<int> ret;
7
8 void dfs(int idx, int clr, std::vector<int> &color, std::vector<std::vector<int>>& grph) {
9     color[idx] = clr;
10    for (auto v : grph[idx]) {
11        if (color[v] == 1){
12            bad_graph = true;
13        }
14        if (color[v] == 0) {
15            dfs(v, 1, color, grph);
16        }
17    }
18    ret.push_back(idx);
19    color[idx] = 2;
20 }
21
22 int main() {
23     int n, m;
24     std::cin >> n >> m;
25
26     std::vector<std::vector<int>> grph(n + 1);
27     std::vector<int> color(n + 1, 0);
28
29     for (int i = 0, g1, g2; i < m; ++i) {
30         std::cin >> g1 >> g2;
31         grph[g1].push_back(g2);
32     }
33
34     for (int i = 1; i <= n; ++i) {
35         if (color[i] == 0) {
36             dfs(i, 1, color, grph);
37         }
38     }
39
40     if (bad_graph) {
41         std::cout << -1;
42     } else {
43         for (int i = 0; i < ret.size(); ++i) {
44             std::cout << ret[i] << " ";
45         }
46         std::cout << "\n";
47     }
48 }
```

```
37
38     if (bad_graph){
        std::cout << -1;
```

[Отправить](#)[Предыдущая](#)[Следующая](#)