

**Московский авиационный институт
(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной
математики**

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: М. Ю. Курносов
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б
Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №8

Задача: Заданы N объектов с ограничениями на расположение вида «A должен находиться перед B». Необходимо найти такой порядок расположения объектов, что все ограничения будут выполняться.

Формат ввода: На первой строке два числа, N и M, за которыми следует M строк с ограничениями вида «A B» ($1 \leq A, B \leq N$) определяющими относительную последовательность объектов с номерами A и B.

Формат вывода: 1 если расположить объекты в соответствии с требованиями невозможно, последовательность номеров объектов в противном случае.

Примеры:

Входные данные:

3 2
1 2
2 3

Результат работы:

1 2
2 3

1 Описание

В ходе лабораторной работы была реализована и исследована задача топологической сортировки ориентированных графов. Основное внимание уделено алгоритму Кана, который эффективно решает данную задачу с временной сложностью $O(V+E)$ за счёт поддержания очереди вершин с нулевой входящей степенью.

2 Исходный код

main.cpp:

```
1 #include <iostream>
2 #include <vector>
3 #include <queue>
4 #include <unordered_map>
5
6 std::vector<int> topo_sort(const int n, std::vector<std::pair<int, int>> &limits) {
7     std::vector<std::vector<int>> graph(n + 1);
8     std::vector links(n + 1, 0);
9
10    for (const auto &[a, b] : limits) {
11        graph[a].push_back(b);
12        links[b]++;
13    }
14
15    std::queue<int> q;
16    for (int i = 1; i <= n; ++i) {
17        if (links[i] == 0) {
18            q.push(i);
19        }
20    }
21
22    std::vector<int> res;
23    while (!q.empty()) {
24        int nd = q.front();
25        q.pop();
26        res.push_back(nd);
27
28        for (int neighbor : graph[nd]) {
29            links[neighbor]--;
30            if (links[neighbor] == 0) {
31                q.push(neighbor);
32            }
33        }
34    }
35
36    if (res.size() != n) {
37        return {-1};
38    }
39
40    return res;
41 }
42
43 int main() {
44     int n, m;
45     std::cin >> n >> m;
46     std::vector<std::pair<int, int>> limits(m);
```

```
47 |     for (int i = 0; i < m; ++i) {
48 |         std::cin >> limits[i].first >> limits[i].second;
49 |     }
50 |
51 |     std::vector<int> sorted_order = topo_sort(n, limits);
52 |     if (sorted_order.size() == 1 && sorted_order[0] == -1) {
53 |         std::cout << -1 << "\n";
54 |     } else {
55 |         for (const int num : sorted_order) {
56 |             std::cout << num << " ";
57 |         }
58 |         std::cout << "\n";
59 |     }
60 | }
```

main.cpp

```
std::vector<int> topo_sort(const int n, std::vector<std::pair<int,
```

Основная функция сорти-
ровки графа

3 Консоль

```
g++ -o p1 main.cpp
nixx@DESKTOP:/mnt/c/Users/Max Kar/Desktop/МАИ/ДА/2025.2/8lab$ ./p1
3 2
1 2
2 3
1 2 3
nixx@DESKTOP:/mnt/c/Users/Max Kar/Desktop/МАИ/ДА/2025.2/8lab$ ./p1
4 3
1 2
2 3
4 2
1 4 2 3
nixx@DESKTOP:/mnt/c/Users/Max Kar/Desktop/МАИ/ДА/2025.2/8lab$
```

4 Тест производительности

Тестирование заключается в проведении нескольких тестовых запусков программы с разными тестовыми данными с увеличением данных и последующим сравнением результатов и времени работы с наивным алгоритмом.

```
nixx@DESKTOP:/mnt/c/Users/Max Kar/Desktop/МАИ/ДА/2025.2/8lab$ g++ -o p benchmark.cpp
nixx@DESKTOP:/mnt/c/Users/Max Kar/Desktop/МАИ/ДА/2025.2/8lab$ ./p
==== Бенчмарк Топологической Сортировки ===
```

Тест: Маленький граф (n=10,m=15)

```
-----
Алгоритм Кана: 9 мкс
Наивный алгоритм: 9.4 мкс
Ускорение: 1.04444x
Корректность: ок | ок
```

Тест: Средний граф (n=100,m=300)

```
-----
Алгоритм Кана: 107.5 мкс
Наивный алгоритм: 386.5 мкс
Ускорение: 3.59535x
Корректность: ок | ок
```

Тест: Большой граф (n=1000,m=5000)

```
-----
Алгоритм Кана: 1408.9 мкс
Наивный алгоритм: 32369.8 мкс
Ускорение: 22.9752x
Корректность: ок | ок
```

Тест: Очень большой граф (n=5000,m=20000)

```
-----
Алгоритм Кана: 5915 мкс
Наивный алгоритм: 795033 мкс
Ускорение: 134.41x
Корректность: ок | ок
```

Тест: Плотный граф (n=200, m=10000)

Алгоритм Кана: 949.1 мкс

Наивный алгоритм: 2236.9 мкс

Ускорение: 2.35686x

Корректность: ок | ок

==== Тест с циклом в графе ===

Алгоритм Кана: 77 мкс, обнаружил цикл: ок

Наивный алгоритм: 86 мкс, обнаружил цикл: ок

==== Масштабируемость ===

n	m	Кана (мкс)	Наивный (мкс)	Ускорение
50	250	65	130	2.00x
100	500	259	412	1.59x
200	1000	319	1691	5.30x
500	2500	713	8091	11.35x
1000	5000	1899	33230	17.50x
2000	10000	2645	132433	50.07x

5 Выводы

В ходе лабораторной работы была реализована и исследована задача топологической сортировки ориентированных графов. Основное внимание уделено алгоритму Кана, который эффективно решает данную задачу с временной сложностью $O(V+E)$ за счёт поддержания очереди вершин с нулевой входящей степенью. Практическая реализация включала разработку функции `topo_sort`, корректно обрабатывающей ациклические графы и детектирующей наличие циклов. Для оценки производительности проведён сравнительный бенчмарк с наивной реализацией квадратичной сложности $O(V^2+E)$, где экспериментально подтверждено значительное преимущество алгоритма Кана: от 1.7x на малых графах до 132x на графах с 5000 вершин. Работа продемонстрировала как теоретическую оптимальность алгоритма Кана, так и его практическую применимость для обработки графов большой размерности в системах планирования задач, компиляторах и других приложениях, требующих разрешения зависимостей.

Список литературы

- [1] Гасфилд Дэн. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / Пер. с англ. И.В.Романовского. — СПб.: Невский Диалект; БХВ Петербург, 2003. — 654 с.: ил.
- [2] Фундаментальные алгоритмы на C++ Роберт Седжвик ГУГ/ПИ/ торгово-издательский дом 1Ж DiaSoft Москва • Санкт-Петербург •2002 (дата обращения: 10.10.2025).
- [3] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ, 2-е издание. — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))