



Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский
университет)» (МГТУ им. Н.Э. Баумана)**

Лабораторная Работа №8 «Графы » Вариант №1

Студент **Шахнович Дмитрий Сергеевич**

Группа **ИУ7-22Б**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент Шахнович Д.С.

Оценка _____

2023 г.

Описание условия задачи

Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Предложить вариант реальной задачи, для решения которой можно использовать разработанную программу. Результат выдать в графической форме.

Задание: Найти все вершины заданного орграфа, недостижимые из заданной его вершины.

Техническое задание

Исходные данные:

Граф, заданный файлом или консольным вводом.

Формат файла или ввода: В начале целое положительное число n — количество вершин в графе.

Далее n строк — названий соответствующих вершин.

Далее идет ввод ребер орграфа в формате «а б», где а — индекс начальной вершины ребра, б — конечной, при этом индексы нумеруются с 0.

При запуске алгоритма поиска указывается индекс начальной вершины.

Выходные данные:

Графическое представление графа, где отмечены вершины, недостижимые из заданной вершины.

Описание задания:

Найти все вершины заданного орграфа, недостижимые из заданной его вершины.

Способы обращения к программе:

Запуск программы через терминал, затем управление программой с помощью меню. Пункты меню:

- 1 — Задать граф;
- 2 — Найти недостижимые вершины поиском в глубину;
- 3 — Найти недостижимые вершины поиском в ширину;
- 4 — Сравнить алгоритмы;
- 0 — Выход.

Аварийные ситуации:

- 1) Ввод несуществующей команды в меню;
Сообщение: «Ошибка: Некорректная команда.»
- 2) Ошибка ввода/вывода;
Сообщение: «Ошибка функций ввода/вывода.»
- 3) Неудачная попытка работы с файлом
Сообщение: «Ошибка при работе с файлом.»
- 4) Неудачная попытка выделения памяти;
Сообщение: «Ошибка выделения памяти.»
- 5) Ввод литералов или чисел вне запрашиваемого диапазона;
Сообщение: «Ошибка: Некорректный формат ввода.»
- 6) Ввод названия несуществующего файла;
Сообщение: «Ошибка: Не удалось использовать файл.»
- 7) Ввод некорректного числа или символа:
Сообщение: «Ошибка: Некорректный формат ввода.»
- 8) Ввод некорректных индексов узлов графа;
Сообщение: «Ошибка: Некорректные границы.»

Описание структур данных

```
/// @brief Узел стека в виде односвязного списка
struct stack_node_t
{
    int data; /// Значение узла
    stack_node_t *next; /// Указатель на следующий(более близкий к
начальному) элемент стека.
};
```

```
/// Узел очереди-списка заявок
struct qnode
{
    int data; /// Часть узла с данными в виде заявок.
    qnode_t* next; /// Указатель на следующий узел.
};

/// Очередь-список
struct list_queue
{
    qnode_t *pin; /// Указатель на входную часть списка.
    qnode_t *pout; /// Указатель на выходную часть списка.
};
```

```
/// @brief Структура узла оргграфа.
struct ornode
{
    size_t id; /// Идентификатор узла.
    char *name; /// Имя узла.
    int mark; /// Метка узла
};

/// @brief Список смежностей оргграфа.
```

```

struct orlist {
    ornode_t *data; /// Информационная часть узла.
    orlist_t *next; /// Указатель на следующий элемент списка.
};

/// @brief Структура оргграфа в виде списков смежностей.
struct orgraph
{
    orlist_t *nodes; /// Массив списков смежных узлов.
    size_t size; /// Количество узлов в графе.
    size_t max_size; /// Максимальное количество узлов.
};

```

Описание алгоритма

Для решения задачи я решил использовать алгоритмы поиска в глубину и поиска в ширину. Данную задачу можно решить, обойдя по разу все достижимые узлы и проверив каждый из их ребер, поэтому алгоритмы обхода идеально подходят для задачи, а так как они имеют одинаковую сложность, то я также решил проверить какой из них будет эффективнее для данной задачи.

В качестве структуры хранения я выбрал списки смежностей, так как на небольшой заполненности графа он работает лучше матрицы с точки зрения обходов (Сложность обхода в списке $O(|V| + |E|)$, в матрице $O(|V|^2)$), а на мой взгляд вопрос достижимости вершин может чаще возникать в именно в малозаполненных графах.

Реальные задачи: Поиск недостижимых мест в схеме авиаперелетов, поиск недостижимых устройств в различных топологиях сети.

Алгоритм поиска в глубину.

1. Начинаем с заданной начальной точки, помечаем её посещенной.
2. В списке смежности берем первую вершину.

3. Заносим начальную вершину в стек.
4. Помечаем найденную вершину посещенной и ищем в списке смежности новую не посещенную вершину.
5. Если удалось найти вершину, то нынешнюю вершину заносим в стек и переходим с найденной в п.4
6. Если не удалось найти вершину, то из стека забираем вершину и с ней переходим к п.4

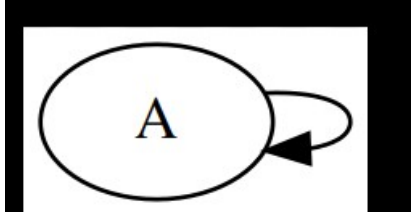
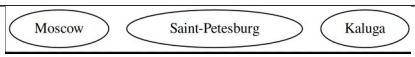
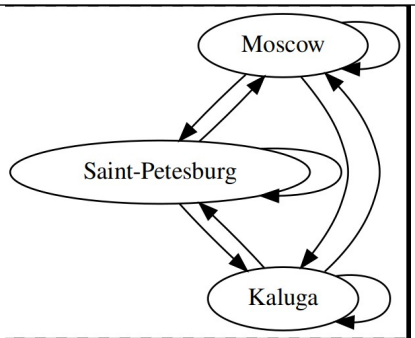
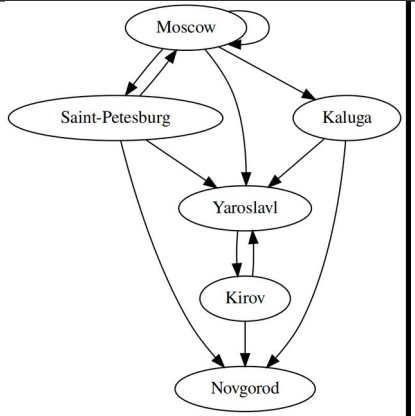
П.4-6 проделывать до тех пор, пока не опустеет стек.

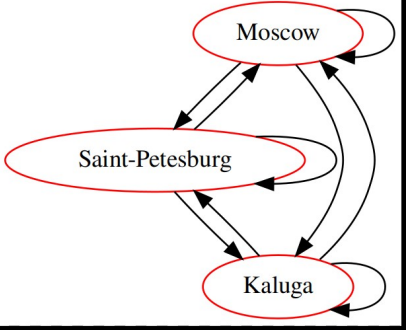

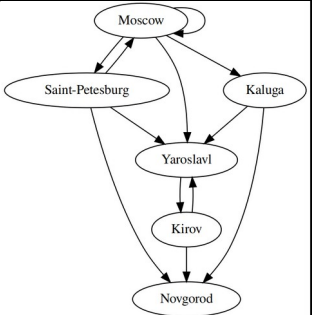
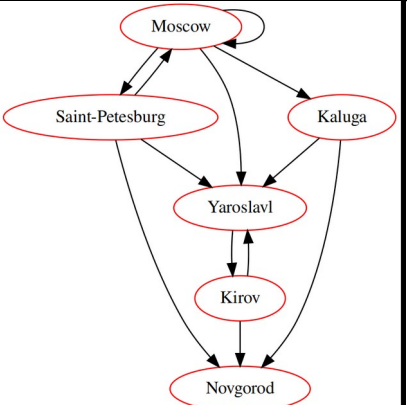
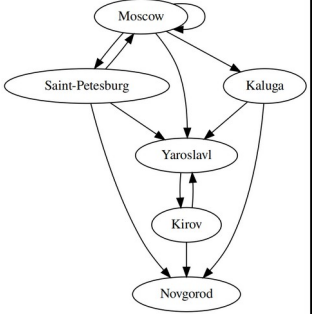
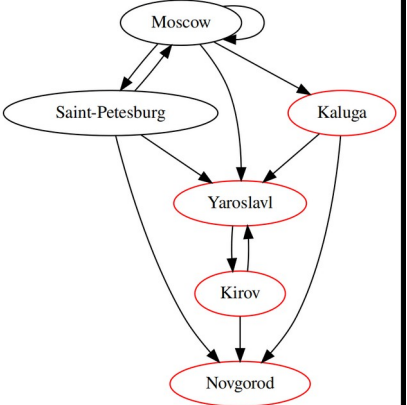
Алгоритм поиска в ширину.

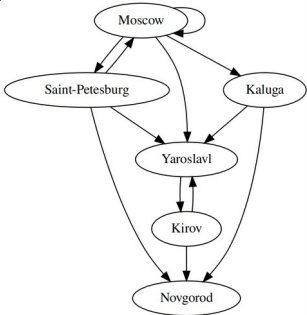
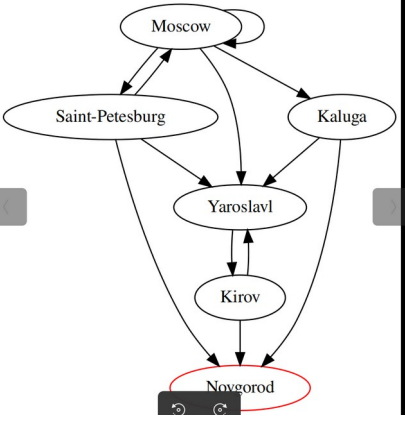
1. Помечаем начальный узел как посещенный и все его вершины из списка смежности заносим в очередь.
 2. Достаём из очереди узел и помечаем его посещенным.
 3. Находим все не посещенные узлы в списке смежности предыдущего узла и помещаем их в очередь, затем возвращаемся к п.2
- П.2-3 проделывать до тех пор, пока не опустеет очередь.

В результате обоих алгоритмов все достижимые вершины будут помечены посещенными, а недостижимые — нет.

Тестовые данные

Позитивные тесты			
№	Описание	Вход	Выход
1	Ввести граф с размером 0	0	Граф пустой
2	Ввести граф с одной вершиной и ребром на себя	1 A 0 0	
3	Ввести граф без связей		
4	Ввести полносвязный граф	3	
5	Ввести обычный граф	6	

6	Поиск в полносвязном графе	0	
7	Поиск в графе без связей	0	
8	Поиск в не полносвязном графе со всеми достижимыми узлами	 Moscow	
9	Поиск в не полносвязном графе, не все узлы достижимы	 Kaluga	

10	Поиск из недостижимой точки	 <p>Novgorod</p>	
Негативные тесты			
1	Ввести некорректный код в меню.	16	Ошибка: Некорректная команда.
2	При запросе числа ввести литерал.	sda	Ошибка: Некорректный формат ввода.
3	Ввести несуществующее имя файла	sdw	Ошибка: Не удалось использовать файл.
4	При вводе графа ввести некорректные индексы узлов	Размер графа: 7 ввод: 8 1	Ошибка: Некорректные границы.

Замеры эффективности

Замеры выполнения поиска проводились 100000 раз для каждого тестового файла с графами. Каждый граф создавался заранее скриптом случайно.

Заполненность графа — процент ребер от их максимального количества, то есть от полносвязного графа.

Замеры умножения по времени

Кол-во элементов.	Заполненность графа, %.	Поиск в глубину, нс	Поиск в ширину, нс	Отношение поиска в ширину к поиску в глубину
5	5	46	56	1.21
	10	51	62	1.21
	15	60	73	1.21
	20	50	70	1.4
	30	109	117	1.07
	50	170	191	1.12
10	5	55	72	1.3
	10	88	104	1.18
	15	174	163	0.93
	20	142	148	1.04
	30	332	444	1.33
	50	386	680	1.76
15	5	88	96	1.09
	10	324	330	1
	15	344	348	1
	20	512	631	1.2
	30	540	852	1.6
	50	619	1287	2.07
20	5	104	109	1
	10	418	410	1
	15	635	837	1.3
	20	630	1068	1.68
	30	742	1422	2
	50	868	2405	3

25	5	161	158	1
	10	687	751	1.1
	15	823	1208	1.5
	20	905	1704	2.1
	30	1005	2435	2.4
	50	1243	3789	3.1
30	5	346	299	0.8
	10	877	1124	1.375
	15	1075	1803	1.8
	20	1115	2643	2.3
	30	1276	3373	2.5
	50	1582	5959	3.6

Как видно в при маленьких размерах графа алгоритмы работают примерно одинаково, однако при большом размере и большой заполненности выигрывает поиск в глубину.

Ответы на вопросы

1. Что такое граф?

Граф — множество вершин и ребер, соединяющих их.

2. Как представляются графы в памяти?

Графы могут представляться матрицами смежности и списками смежности.

Матрица смежности — матрица, в каждой ячейке которой лежит информация о существовании ребра между соответствующими узлами.

Список смежности — список, в котором хранятся все вершины, смежные данным. Такие списки составляются по всем вершинам и дают полную картину от графа.

3. Какие операции возможны над графами?

Основные операции — поиск кратчайших путей в графе, поиск эйлеровых путей, поиск гамильтоновых путей.

4. Какие способы обхода графов существуют?

Обходы графа в ширину и глубину.

5. Где используются графовые структуры?

Графовые структуры используются для представления различных путевых структур, таких как дороги или рейсы авиаперелетов. Также они часто используются в компьютерных сетях.

6. Какие пути в графе Вы знаете?

Простой — Все вершины попарно различны

Элементарный — Все вершины, кроме начала и конца попарно различны

Эйлеров — Проходит по всем ребрам ровно один раз

Гамильтонов — Проходит по всем узлам ровно один раз

7. Что такое каркасы графа?

Каркас — Остовое дерево — связный подграф данного графа, в котором содержатся все вершины данного графа, и не содержащий циклов.

Выводы

В большинстве современных системах в тех или иных задачах возникает необходимость представлять данные в виде графов, поэтому для любого программиста важно хорошо понимать их устройство и алгоритмы, а также грамотно использовать их в задачах.