Отчет

Московец Наталии
Группа: ИУ7-31

Лабораторная работа №8
Графы

Цель работы: реализовать алгоритмы обработки графовых структур: поиск различных путей, проверку связности, построение остовых деревьев минимальной стоимости.

Задание (Вариант 1):

Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Ввод данных – на усмотрение программиста. Результат выдать в графической форме.

Найти все вершины заданного орграфа, недостижимые из заданной его вершины

Требования к входным данным:

Программа считывает данные о графе из консоли или из текстового файла. Граф задается количеством вершин и списком ребер. Ребро задается парой целых чисел (вершина из которой ведет дуга).

Выходные данные:

По требованию пользователя программа печатает список вершин, недостижимых из введенной, а также сохраняет граф в формате, который потом с помощью программы graphviz можно визуализировать и сохранить в виде картинки в формате .png. Примеры см. Интерфейс программы.

Интерфейс программы:

```
****************************************************************
Выберите одно из следующих действий:
0: Загрузить граф из файла
1: Задать граф
2: Вывод вершин, недостижимых из исходной
3: Визуализация графа
4: Закончить работу
```

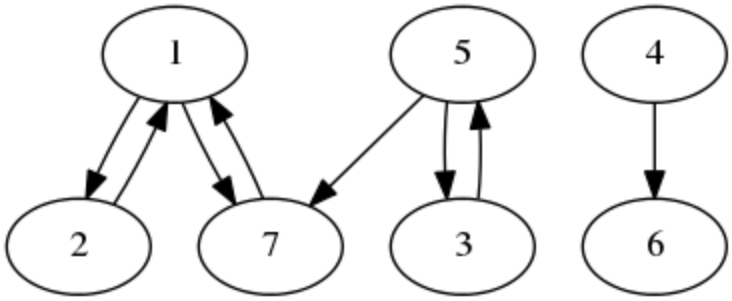
0 - Задание графа из файла data.txt. Файл содержит количество вершин и список дуг.

```
7
3 5
5 3
4 6
7 1
1 7
1 2
2 1
5 7
```

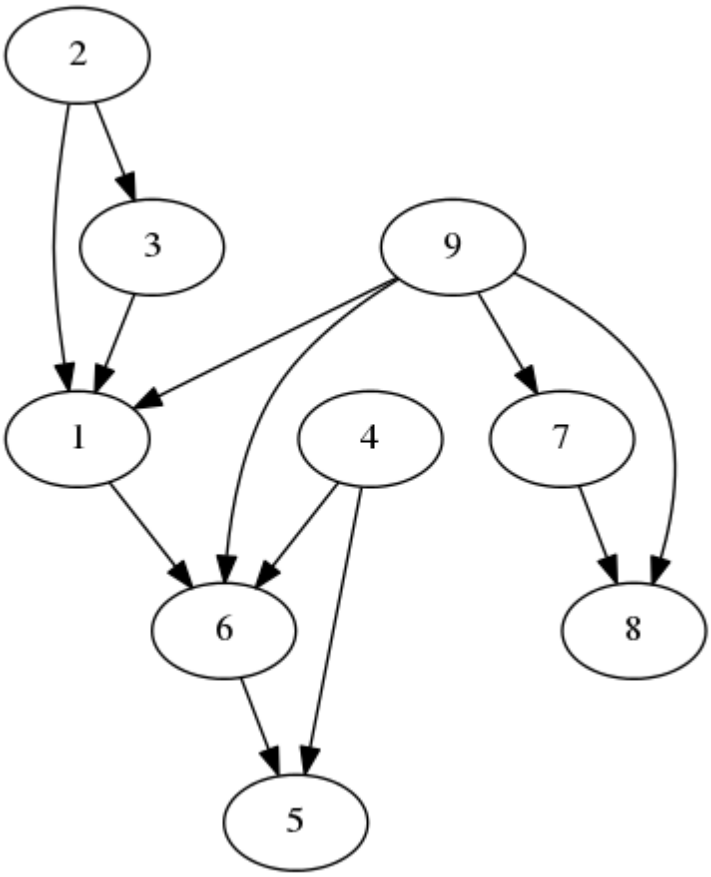
1 - Пользователь вводит номер вершин (от 1 до n). Программа печатает список вершин, недостижимых из введенной.
3 - Программа сохраняет граф в файле graph.gv. Чтобы сгенерировать картинку, необходимо в файле с проектом выполнить команду

dot -Tpng graph.gv -o temp.png

В temp.png находится отображение графа.



4 - Закончить работу



Пример

Из вершины 8 недостижимы все вершины. Программа выведет список 1 2 3 4 5 6 7 9
Из вершины 2 недостижимы вершины 4 7 8 9
Из вершины 9 недостижимы 2 3 4

Обращение к программе:

Через консоль

Внутренние структура данных:

Граф представлен в виде матрицы смежности размера n*n, где в (i,j) ячейке хранится 0, если дуги между вершинами нет, и 1 в противном случае.
Матрица смежности является более удобным способом хранения данных для их

обработки алгоритмами обхода. Недостатком выбранной реализации является большое количество требуемой памяти, а также необходимость для каждой вершины пройтись по всей строке данной матрицы. Так как в данной лабораторной исходные данные небольшие (n <= 1000), то данную структуру данных использовать можно.

Алгоритм поиска вершин

Для реализации поставленной задачи необходимо выполнить обход графа из указанной вершины и вывести пользователю все вершины, которые в результате обхода остались не посещенными.
В качестве алгоритма обхода был выбран алгоритм поиска в глубину. Так как, для его реализации не нужно использовать дополнительно написанную очередь, как это необходимо при поиске в ширину, а в качестве стека можно использовать стек вызовов рекурсивной функции, то реализация данного алгоритма более проста.

Вывод

В данной лабораторной работе был реализован обход вершин графа для поиска всех вершин, не достижимых из заданной. В качестве основного алгоритма был выбран обход в глубину. В программе граф представляется в виде матрицы смежности.

Теоретическая часть:

1. Что такое граф?

Граф – конечное множество вершин и соединяющих их рёбер; G = <V, E>. Если пары E (ребра) имеют направление, то граф называется направленным; если ребро имеет вес, то граф называется взвешенным.

2. Как представляются графы в памяти?

Существуют различные методы представления графов в программе.

Матрица смежности B(n*n) – элемент b[i,j]=1, если существует ребро, связывающее вершины i и j, и =0, если ребра не существует.

Список смежностей – содержит для каждой вершины из множества вершин V список тех вершин, которые непосредственно связаны с ней. Входы в списки смежностей могут храниться в отдельной таблице, либо же каждая вершина может хранить свой список смежностей.

3. Какие операции возможны над графами?

Основные операции над графами: обход вершин и поиск различных путей: кратчайшего пути от вершины к вершине; кратчайшего пути от вершины ко всем остальным; кратчайших путей от каждой вершины к каждой; поиск эйлерова пути и гамильтонова пути, если таковые есть в графе.

4. Какие способы обхода графов существуют?

Один из основных методов проектирования графовых алгоритмов – поиск в глубину. Начиная с некоторой вершины v0, ищется ближайшая смежная ей вершина v, для которой в свою очередь осуществляется поиск в глубину до тех пор, пока не встретится ранее просмотренная вершина, или не закончится список смежности вершины v (то есть вершина полностью обработана). Если нет новых вершин, смежных с v, то вершина v считается использованной, идет возврат в вершину, из которой попали в вершину v, и процесс продолжается до тех пор, пока не получим v = v0. При просмотре используется стек.

Поиск в ширину – обработка вершины V осуществляется путём просмотра сразу всех «новых» соседей этой вершины, которые последоватеьно заносятся в очередь просмотра.

Для поиска кратчайших путей используются алгоритмы Дейкстры, Беллмана-Форда, Флойда-Уоршалла.

5. Где используются графовые структуры?

Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи, необязательно иерархические. Наиболее распространенным является использование графов при решении различных задач о путях, будь то построение коммуникационных линий между городами или прокладка маршрута на игровом поле.

6. Какие пути в графе Вы знаете?

Путь в графе, проходящий через каждое ребро ровно один раз, называется эйлеровым путём; путь может проходить по некоторым вершинам несколько раз – в этом случае он является непростым.

Путь, проходящий через каждую вершину ровно один раз, называется гамильтоновым путём.

Как эйлеров, так и гамильтонов путь могут не существовать в некоторых графах.

7. Что такое каркасы графа?

Каркас графа – дерево, в которое входят все вершины графа, и некоторые (не обязательно все) его рёбра.

Для построения каркасов графа используются алгоритмы Крускала и Прима.