**Лабораторная работа №8**

Янова Даниэлла, ИУ7-33

Графы

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

**Задание (Вариант 9)​:**

Задана система двусторонних дорог, где для любой пары городов есть соединяющий их путь. Найти город с минимальной суммой расстояний до остальных городов.

**Требования к входным данным:**

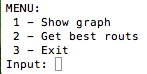
Программа считывает данные о графе из консоли или из текстового файла. Граф задается матрицей с нулевой главной линией.

**Выходные данные:**

Программа печатает на консоль матрицу входного графа и матрицу кратчайших путей.

**Обращение к программе:** Через консоль

**Интерфейс:**



**Внутренние структуры 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.

typedef struct matrix\_t

{

unsigned int n; // The number of columns

float \*\*data;

} matr;

**Алгоритм поиска кратчайшего пути**:

Для поиска кратчайших путей между всеми вершинами используется алгоритм Флойда-Уоршалла. По алгоритму Флойда-Уоршалла сначала ищется кратчайший путь от одной вершины ко всем вершинам, доступным из нее, затем проводятся те же действия, но пытаясь пройти от этой вершины ко всем доступным из нее, проходя каждый раз через новую вершину (сначала через первую, затем – через вторую и т.д.). Таким образом обрабатываются все вершины.

**Вывод:**

В данной лабораторной работе был реализован обход вершин графа для поиска города с минимальной суммой расстояний до остальных городов. 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 Е (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.