Лабораторная работа №8

1. Задание:

Задана система двусторонних дорог. Для каждой пары городов найти длину кратчайшего пути между ними. Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Ввод данных – на усмотрение программиста.

2. ТЗ:

Цель работы – получить навыки работы с графами.

Входные данные.

На вход программа получает файл. первая строчка файла - количество вершин графа. Далее идет матрица - вес пути между i-той и j-той вершинами. По умолчанию отсутствие дороги (INF) принято за число 50.

Выходные данные.

В результате работы программа выводит матрицу смежности введенного графа и полученный после преобразований граф.

3. СД:

В качестве структуры для работы с данной задачей была выбрана матрица смежности, так как она обеспечивает удобство применения выбранного для решения задачи алгоритма - алгоритма Флойда-Уоршелла (подробнее в #4).

Матрица смежности:

```
int **matrix = (int**)malloc(sizeof(int)*numberOfVert);
    for(int i = 0; i < numberOfVert; i++)
        matrix[i] = (int *) malloc(sizeof(int) * numberOfVert);
```

4. Алгоритм:

Для решения данной задачи был использован алгоритм Флойда-Уоршелла - динамический алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. Этот алгоритм лучше всего подходит для данного задания.

Основная идея алгоритма:

```
for k = 1 to n
  for i = 1 to n
    for j = 1 to n
      graph[i][j] = min(graph[i][j], graph[i][k] + graph[k][j])
```

5. Тесты:

Тест №1.	Тест №2.
Исходный граф: 0 1 2 3 --- 0| 0 10 1 – 1| 10 0 – 5 2| 1 – 0 2 3| – 5 2 0 Граф кратчайших путей: 0 1 2 3 --- 0| 0 8 1 3 1| 8 0 7 5 2| 1 7 0 2 3| 3 5 2 0	Исходный граф: 0 1 2 3 --- 0| 0 – 1 – 1| – 0 – – 2| 1 – 0 2 3| – – 2 0 Граф кратчайших путей: 0 1 2 3 --- 0| 0 – 1 3 1| – 0 – – 2| 1 – 0 2 3| 3 – 2 0

6. Выводы:

В результате работы были основаны методы работы с графами. Был реализован алгоритм Флойда-Уоршелла, позволяющий найти кратчайшие расстояния между всеми вершинами взвешенного ориентированного графа. Данный алгоритм имеет кубическую сложность: $O(n^3)$.

7. Контрольные вопросы:

1) Что такое граф?

- Граф - абстрактный математический объект, представляющий собой множество вершин графа и набор рёбер, то есть соединений между парами вершин.

2) Как представляются графы в памяти?

- Графы в памяти могут представляться различным способом. Один из видов представления графов – это матрица смежности B(n*n); В этой матрице элемент b[i,j]=1, если ребро, связывающее вершины Vi и Vj существует и b[i,j]=0, если ребра нет. У неориентированных графов матрица смежности всегда симметрична. Во многих

случаях удобнее представлять граф в виде так называемого списка смежностей. Список смежностей содержит для каждой вершины из множества вершин V список тех вершин, которые непосредственно связаны с этой вершиной. Каждый элемент (ZAP[u]) списка смежностей является записью, содержащей данную вершину и указатель на следующую запись в списке (для последней записи в списке этот указатель – пустой). Входы в списки смежностей для каждой вершины графа хранятся в таблице (массиве) (BEG [u])

3) Какие операции возможны над графами?

- поиск кратчайшего пути от одной вершины к другой (если он есть), поиск кратчайшего пути от одной вершины ко всем другим, поиск кратчайших путей между всеми вершинами, поиск эйлерова пути (если он есть), поиск гамильтонова пути (если он есть).

4) Какие способы обхода графов существуют?

- Поиск в глубину и поиск в ширину.

5) Где используются графовые структуры?

- Например, существует граафовая база данных — разновидность баз данных с реализацией сетевой модели в виде графа и его обобщений.

6) Какие пути в графе Вы знаете?

- Произвольный путь в графе, проходящий через каждое ребро графа точно один раз, называется эйлеровым путем. При этом, если по некоторым вершинам путь проходит неоднократно, то он является непростым. Если путь замкнут, то имеем эйлеров цикл. Для существования эйлерова пути в связном графе необходимо и достаточно, чтобы граф содержал не более двух вершин нечетной степени. Путь в графе, проходящий в точности один раз через каждую вершину графа (а не каждое ребро) и соответствующий цикл называются гамильтоновыми и существуют не для каждого графа, как и эйлеров путь. В отличие от эйлеровых путей неизвестно ни одного простого необходимого и достаточного условия для существования гамильтоновых путей. Неизвестен даже алгоритм полиномиальной сложности, проверяющий существование гамильтонова пути в произвольном графе. Проблема существования гамильтонова пути принадлежит к классу так называемых NP-полных задач. Также существует понятие кратчайшего пути.

7) Что такое каркасы графа?

- При использовании алгоритмов DFS и BFS графы обходят разными способами, получая при этом некоторые подграфы, которые имеют специфические названия: каркасы, остовы или стягивающие деревья.