**Обработка графовых структур**

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

Задана система двусторонних дорог. Для каждой пары городов найти длину кратчайшего пути между ними.

**Входные данные**

N- размерность таблицы расстояний(натуральное число не более 100).Таблица расстояний между между города(натуральные числа меньшие 1000). В случае отсутствия дороги между городами, расстояние между ними принимается равным 1000.

**Выходные данные**

Таблица кратчайших расстояний между городами, графические интерпретация этой таблицы.

**Допущения**:

Считается ,что таблица расстояний между города введена верно и то, что из любого города можно попасть в любой.

**Функции и алгоритмы**

Для поиска пути был выбран алгоритм Флойда — Уоршелла, служащий для нахождения кратчайшего пути между всеми вершинами взвешенного графа, то бишь как раз для решения нашей задачи. На вход этот граф получает массив с длинами ребер графа, где элемент :

- длина ребра (i,j).

**Структуры данных**

int d[N][N] – таблица длин ребер

int W[N][N] – таблица кратчайших путей

**Алгоритм**

На каждом шаге алгоритм генерирует матрицу W. Матрица W содержит длины кратчайших путей между всеми вершинами графа. Перед работой алгоритма матрица W заполняется длинами рёбер графа (или запредельно большим M(1000), если ребра нет).

**for** (**int** i=0;i<N;i++)

**for** (**int** j=0;j<N;j++)

W[i][j]=d[i][j];

**for** (**int** k=0;k<N;k++)

**for** (**int** i=0;i<N;i++)

**for** (**int** j=0;j<N;j++)

W[i][j] = min(W[i][j], W[i][k] + W[k][j]);

**Сложность алгоритма**

Три вложенных цикла содержат операцию, исполняемую за константное время. O(1) = O(n^3), то есть алгоритм имеет кубическую сложность.