Отчет по лабораторной работе № 7 по дисциплине

“Типы и структуры данных”

Сушина А.Д. ИУ7-31б

Работа № 7

Графы

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

# Описание условия задачи

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

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

# Описание ТЗ

### Общая концепция системы

Программа выполняет считывание и обработку неориентированного графа. Производит поиск длин минимальных путей от всех вершин до остальных вершин. Выводит считанный граф, матрицу путей, а также минимальную сумму расстояний до остальных городов.

### Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

* считать данные о графе из файла
* нарисовать граф
* подсчитать все расстояния от одной вершины до остальных
* найти минимальную сумму расстояний
* вывод пояснений при выводе результата
* вывод времени работы программы
* отрабатывать неверные входные данные

На **вход** программа получает имя файла, из которого будет происходить считывание данных о графе.

**Выход** должен быть представлен в виде: номер вершины и наименьшая сумма расстояний. Также программа должна вывести данные о времени выполнения алгоритма.

Программа должна выдавать корректный ответ при вводе любых данных. Если произошла ошибка ввода, программа должна сообщить об этом.

## Аварийные ситуации

* Невозможно открыть файл
  + Программа выведет “can’t open file”
* Ошибка считывания имени файла
  + Программа выведет “input error”
* Ошибка при считывании данных из файла
  + Программа выведет ”error while reading”
* Между какими-то вершинами не существуют пути
  + Программа выведет”не все вершины достижимы”

## Способ обращения к программе

Программа представляет собой файл app.exe. Запускается в консоли. Для запуска достаточно команды ./app.exe. Если файл отсутствует можно собрать его с помощью утилиты make.

# Описание структур данных

Для работы с графом используем матрицу дуг. Память под матрицу выделяется динамически. Данный способ был выбран для удобства применения метода Флоида-Уоршела

int \*\*matrix; - матрица дуг, затем матрица коротчайших путей

int n; - длина матрицы

FILE \*f; - файл

int min = 0; - минимальная сумма

int ind = 1; - индекс вершины

int sum; - сумма текущей вершины

# Описание алгоритма

Программа в целом:

1. Ввести имя файла
2. Открыть файл
3. Считать данные из файла
4. Записать считанные данные в файл для графического представления графа
5. Вывести считанный граф
6. Применить алгоритм Флоида-Уоршела
7. Вывести полученную матрицу
8. Проверить все ли пути существуют
9. Найти минимальную сумму путей
10. Вывести ответ

Алгоритм Флоида-Уоршела

* для каждой вершины К
  + Для каждой строки
    - Для каждого элемента строки
      * Если существует путь в элемент через вершину К
      * И он меньше значения элемента
        + значение элемента = путь через вершину

# Тесты

1.Несуществующий файл

Вход: ггг\\\

Вывод: can’t open file

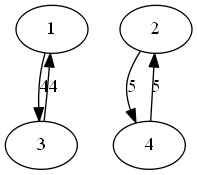
2. Некорректные данные в файле

Вход: -5 -10

Вывод: error while reading

3. Граф, в котором не все пути существуют.

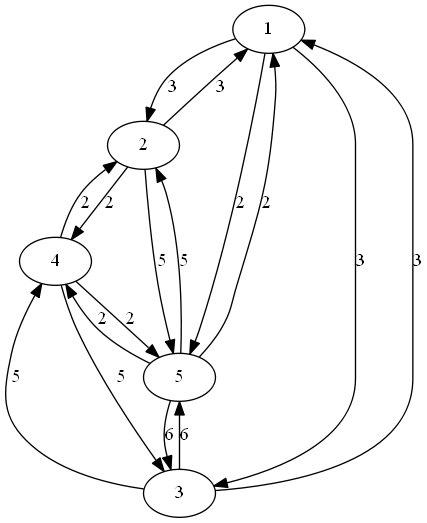
Вход:



Вывод: Не все вершины достижимы

4. Любой граф, где все пути существуют.

Вход:

Выход : 

Программа также выведет таблицу минимальных путей и сумма для каждой вершины для наглядности:

1 2 3 4 5

1 - 0 3 3 4 2

2 - 3 0 6 2 4

3 - 3 6 0 5 5

4 - 4 2 5 0 2

5 - 2 4 5 2 0

sum 1 = 12

sum 2 = 15

sum 3 = 19

sum 4 = 13

sum 5 = 13

**Minimum is 1 string. Sum is 12**

# Выводы по проделанной работе

Была реализована программа, обработки графа и нахождения минимальной суммы путей от одной вершины графа к другой.

Для реализации программы был использован алгоритм Флоида-Уоршела, так как для поиска минимальной суммы путей придется считать минимальный путь для каждой пары вершин. Алгоритм Флоида-Уоршела был разработан именно для решения этой задачи и будет эффективен в данном случае. Сам граф хранится как матрица дуг. Эта реализация была выбрана для удобства реализации алгоритма Флоида-Уоршела, а также для удобства получения матрица коротчайших путей, которая по размерности совпадает с матрицей дуг.

# Ответы на вопросы

1. Что такое граф?

Граф – это конечное множество вершин и ребер, соединяющих их, т. е.: G = < V,E >, где V – конечное непустое множество вершин; Е – множество ребер (пар вершин).

2. Как представляются графы в памяти?

Графы могут представляться как матрица смежностей и как список смежностей.

В матрице элемент b[i,j]=1, если ребро, связывающее вершины Vi и Vj существует и b[i,j]=0, если ребра нет.

Список смежностей содержит для каждой вершины из множества вершин V список тех вершин, которые непосредственно связаны с этой вершин

3. Какие операции возможны над графами?

1. поиск кратчайшего пути от одной вершины к другой (если он есть);
2. поиск кратчайшего пути от одной вершины ко всем другим;
3. поиск кратчайших путей между всеми вершинами;
4. поиск эйлерова пути (если он есть);
5. поиск гамильтонова пути (если он есть).

4. Какие способы обхода графов существуют?

Поиск в глубину и поиск в ширину.

5. Где используются графовые структуры?

Графовые структуры могут использоваться в самых разных областях. В программировании они используются для представление деревьев.

Так же применяется в логистике(при расчете кратчайших маршрутов), в электротехнике.

6. Какие пути в графе Вы знаете?

Гамильтонов путь - путь, проходящий через все вершины

Эйлеров путь - путь, проходящий через все ребра

7. Что такое каркасы графа?

Каркас графа - это дерево, в которое входят все вершины графа и некоторые (необязательно все) его ребра.