**Sandesh Ghimire**

**CSCI 301 section 3**

**Computer Science 2**

**Project 12**

**10th December**

**Introduction**

Floyd Algorithm is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weight. It basically finds the length of shortest paths between all pairs of vertices. It iteratively builds an array whose entries are the shortest path lengths. In this project we ask user for two vertices and display the output with the shortest path between those two vertices.

**Data Structures:**

The data structures used in this project is Array. We mostly used 2D array to do this program. 2D array is used to make matrix that holds the weight of the graph. We used file stream to open a .txt file and extract the data from there. We also used matrix to keep the weights and vertex of the graph.

**Functions:**

The program contains only one file i.e. main.cpp

*Void Floyd()* : This function is a heart of floyd algorithm. It carries out the whole algorithm. It takes two parameters i.e. a matrix and an integer list. It initializes all the lists and matrix first and starts finding the shortest path. After calculating shortest path it asked users to input any two vertices inside a post-condition loop. Then calls a function called *path.*

*Path():* This function takes three parameter one matrix and two integer. The two integers are the value entered by the user. It basically displays the shortest path from entered two vertices by the user.

**The Main Program:**

The mainfunction asks the input to the user first. It asks for the file name. After that the program reads the file name “vertices.txt”. It takes out the vertex and their weight and store them to a variable. After doing all that the main function calls the floyd() function to perform Floyd algorithm.

**Code**

/\*Floyd Algorithm is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weight.

It basically finds the length of shortest paths between all pairs of vertices.

It iteratively builds an array whose entries are the shortest path lengths.

In this project we ask user for two vertices and display the output with the shortest path between those two vertices.\*/

#include <iostream>

#include<iomanip>

#include<cstdlib>

#include <cmath>

#include <fstream>

using namespace std;

void floyd (int g[4][100],int a[4][100]);//floyd algorithm

void path(int p[4][100],int,int);//to display shortest path

void floyd (int g[4][100],int a[4][100])

{

int p[4][100];//matrix for the shortest path

int i,j,k;//declaring variables

int dat;

int inp1=0,inp2=0;

//Initialization

for (i = 1; i<5; i++) { //starting the loop

for (j = 1; j<5; j++) {

a[i][j]=g[i][j];

}

}

//displaying the raw data of matrix

for (i = 1; i<5; ++i) { //starting the loop

for (j = 1; j<5; ++j) {

cout<<a[i][j]<<" ";

}

cout<<endl;

}

cout<<"----------------------------------------------------"<<endl;

//the three major loops of floyd algorithm

for (k=1;k<5;++k){

for (i=1;i<5;++i){

for (j=1;j<5;++j)

{

if(a[i][k]+a[k][j]<a[i][j])

{

a[i][j]=a[i][k]+a[k][j];

p[i][j]=k;

}

}

}

}

//displaying the matrix after computation

cout<<"The shortest path are: \n"<<endl;

for (i = 1; i<5; i++) { //starting the loop

for (j = 1; j<5; j++) {

cout<<a[i][j]<<" ";

}

cout<<endl;

}

do{

cout<<"Enter two vertex numbers: "; //asking input to the user

cin>>inp1;

cin>>inp2;

path(p,inp1,inp2);//calling path function

}while (inp1 != -1);

}

void path(int p[4][100],int i, int j){ //path is a recursive function

int k;

k=p[i][j];

if (k != 1000)//pre-condition

{

path(p,i,k);

cout<<k<<' ';//output the data

path(p,k,j);

}

}

int main(){//The main function

int i, j;

string inp;

int matt[4][100];//matrix

int a[4][100];

int data;

ifstream myfile;

cout<<"ENter the file name: ";//asking input to the user

cin>>inp;

//initializing

for (i = 0; i<5; i++) { //starting the loop

for (j = 0; j<5; j++) {

matt[i][j] = -1;

a[i][j]=-1;

}

}

myfile.open ("vertex.txt");//opening the file

if (!myfile) {//display error message

cout << "Unable to open file";

exit(1); // terminate with error

}

i=0;

while (!myfile.eof())//starting loop if there is no error

{

i=i+1;

for(int j=1;j<5;j++){

myfile >> data;//loading data from txt file

matt[i][j] = data;//keeping the weight in matrix

}

}

myfile.close();//closing the file

floyd(matt,a);//calling the function to evaulate floyd

return 0;

}

**User Document**

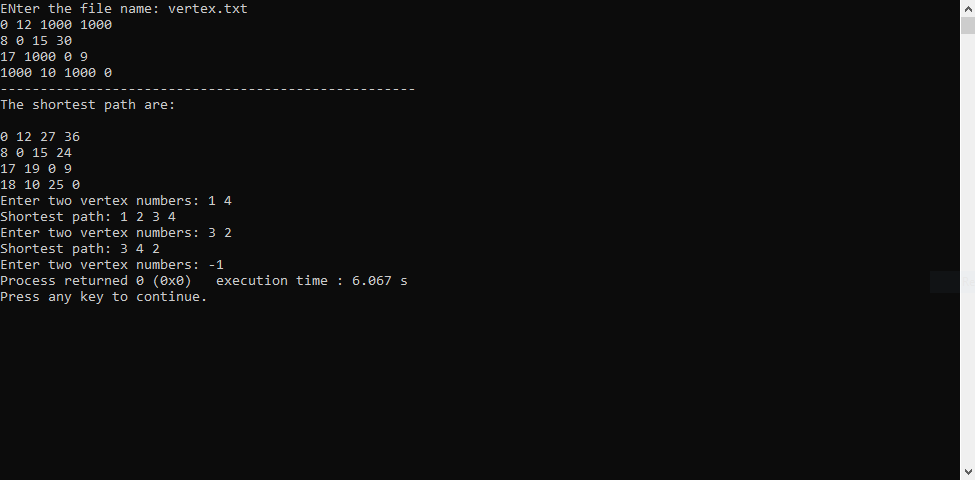
The program can be used by the user by entering the number of vertices they want.

To compile the program simply enter:

*g++ -o main main.cpp*

To run the program, enter. */main,* then you can input the file name. After entering the file name, the program will ask for two vertices. Eventually, it displays the output.

**Testing**



**Summary**

In this project, we have implemented an algorithm that finds the shortest of an undirected edge-weighted graph that is called The Floyd Algorithm. The program reads an input from the user that determines which file to use. It also asks user to enter two vertices and displays the shortest path between them including the path (vertices between them with smallest weight). The problems and their solutions involving Floyd Algorithm that we learnt in class helped a to solve the given project. The Floyd algorithm looks short and easy but when dealing with large amount of data it may take so much time because we used three nested for loop to do it which makes its big-O n^3.