Министерство образования и науки Российской Федерации

Севастопольский Государственный Университет

Кафедра ИС

ОТЧЕТ

по лабораторной работе №5

Поиск кратчайших путей на графах

Выполнил: ст. гр. ИТб-11

Куркчи А.Э.

Проверил:

Шишкевич В.Е.

Севастополь

2015

# 1. Цель

Целью работы является изучение алгоритмов поиска кратчайших путей на графах на примере метода динамического программирования.

2. Постановка задач

Сравнить алгоритмы поиска кратчайшего пути в графе: алгоритм Дейкстры и метод динамического программирования.

3. Текст программы

#include <iostream>

#include <iomanip>

#include <algorithm>

#include <chrono>

#include <climits>

**using** **namespace** std**;**

const int INF **=** 1000000000**;**

bool used**[**100**]** **=** **{**0**};**

int top**[**100**]** **=** **{**0**};**

int g**[**100**][**100**]** **=** **{**0**};**

int n**;**

int l**;**

int s**;**

int f**;**

int d**[**100**]** **=** **{**0**};**

void dfs**(**int v**)** **{**

**if(**used**[**v**])**

**return;**

used**[**v**]** **=** **true;**

**for(**int to **=** 0**;** to **<** n**;** to**++)**

**if(**g**[**v**][**to**])**

dfs**(**to**);**

top**[**l**++]** **=** v**;**

**}**

void topSort**()** **{**

l **=** 0**;**

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

dfs**(**i**);**

reverse**(**top**,** top**+**l**);**

**}**

void solve**()** **{**

int i**,** j**;**

**for(**i **=** 0**;** i **<** n**;** i**++)**

d**[**i**]** **=** INF**;**

d**[**s**]** **=** 0**;**

**for(**i **=** 1**;** i **<** n**;** i**++)**

**for(**j **=** 0**;** j **<** i**;** j**++)**

**if(**g**[**top**[**j**]][**top**[**i**]])**

d**[**top**[**i**]]** **=** min**(**d**[**top**[**i**]],** d**[**top**[**j**]]** **+** g**[**top**[**j**]][**top**[**i**]]);**

**}**

void djeikstra**()** **{**

int i**,**j**,**

v**,**

to**,**

len**;**

**for(**i **=** 0**;** i **<** n**;** i**++)**

d**[**i**]** **=** INF**;**

d**[**s**]** **=** 0**;**

**for(**i **=** 0**;** i **<** n**;** i**++)** **{**

v **=** **-**1**;**

**for(**j **=** 0**;** j **<** n**;** j**++)**

**if(!**used**[**j**]** **&&** **(**v **==** **-**1 **||** d**[**j**]** **<** d**[**v**]))**

v **=** j**;**

used**[**v**]** **=** **true;**

**for(**to **=** 0**;** to **<** n**;** to**++)** **{**

**if(**g**[**v**][**to**])** **{**

len **=** g**[**v**][**to**];**

**if(**d**[**v**]** **+** len **<** d**[**to**])** **{**

d**[**to**]** **=** d**[**v**]** **+** len**;**

**}**

**}**

**}**

**}**

**}**

int main**(**int argc**,**char**\*\*** argv**)** **{**

n **=** 10**;**

cout **<<** "Input start vertex:" **<<** endl**;**

cin **>>** s**;**

cout **<<** "Input finish vertex:" **<<** endl**;**

cin **>>** f**;**

cout **<<** "Input matrix 10x10:" **<<** endl**;**

**for(**int i**=**0**;**i**<**n**;**i**++)** **{**

**for(**int j**=**0**;**j**<**n**;**j**++)** **{**

cin **>>** g**[**i**][**j**];**

**}**

**}**

auto start\_time **=** chrono**::**high\_resolution\_clock**::**now**();**

auto end\_time **=** chrono**::**high\_resolution\_clock**::**now**();**

long long time**;**

start\_time **=** chrono**::**high\_resolution\_clock**::**now**();**

**for(**int i**=**0**;**i**<**1000000**;**i**++)** **{**

**for(**int j**=**0**;**j**<**100**;**j**++)**

used**[**j**]** **=** **false;**

djeikstra**();**

**}**

end\_time **=** chrono**::**high\_resolution\_clock**::**now**();**

time **=** chrono**::**duration\_cast**<**chrono**::**milliseconds**>(**end\_time**-**start\_time**).**count**();**

cout **<<** "By djeikstra: " **<<** d**[**f**]** **<<** endl**;**

cout **<<** "Time: " **<<** time **<<** endl**;**

start\_time **=** chrono**::**high\_resolution\_clock**::**now**();**

**for(**int i**=**0**;**i**<**1000000**;**i**++)** **{**

**for(**int j**=**0**;**j**<**100**;**j**++)**

used**[**j**]** **=** **false;**

topSort**();**

solve**();**

**}**

end\_time **=** chrono**::**high\_resolution\_clock**::**now**();**

time **=** chrono**::**duration\_cast**<**chrono**::**milliseconds**>(**end\_time**-**start\_time**).**count**();**

cout **<<** "By DP: " **<<** d**[**f**]** **<<** endl**;**

cout **<<** "Time: " **<<** time **<<** endl**;**

**return** 0**;**

**}**

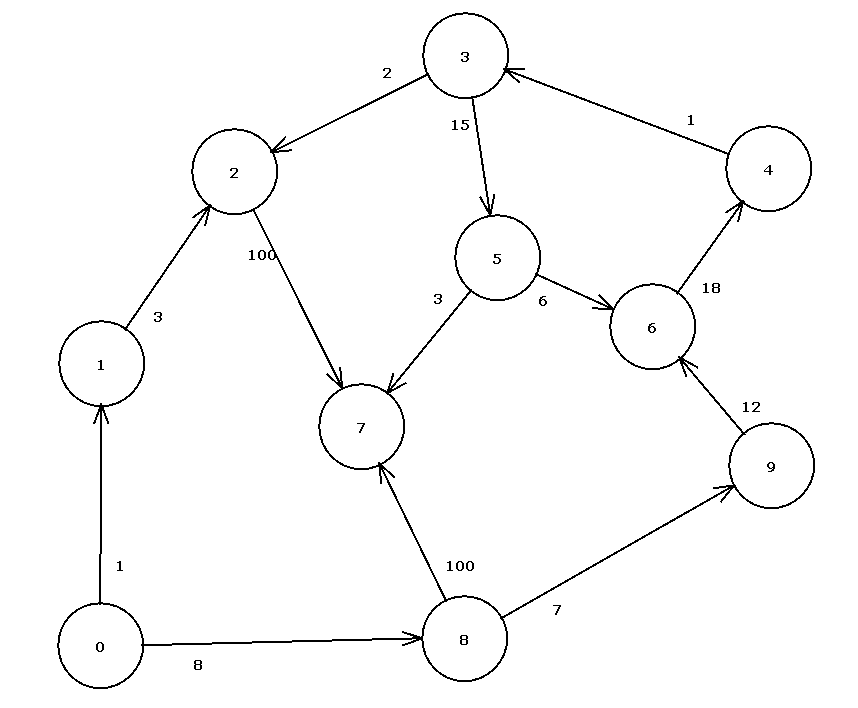
4. Результат

Результаты, полученные в ходе выполнения программы, отображены таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритм | Кол-во прогонов | Кол-во вершин | Время |
| Дейкстры | 1000000 | 10 | 727 |
| Динамическое программирование | 799 |

Таблица 1

Граф представлен на рисунке 2.

Рисунок 2 – Использованный граф

Вывод

В ходе лабораторной работы были изучены алгоритмы поиска кратчайшего пути на графе на примере алгоритма Дейкстры и метода динамического программирования.