Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

Звіт

про виконання лабораторної роботи №8

з дисципліни «Проектний практикум»

|  |  |
| --- | --- |
| Перевірив:  Асистент кафедри ПЗАС  Півень О.Б. | Виконав:  студент 3-го курсу  групи ПЗ-1644  Смірнов І.В. |

Черкаси 2017

**Тема роботи:** Максимізація потоку в мережі методом Форда-Фалкерсона.

**Мета роботи:** Ознайомлення з методами максимізації потоку даних в обчислювальній системі.

**Теоретичні відомості:**

Алгоритм Форда - Фалкерсона вирішує завдання знаходження максимального потоку в транспортній мережі.

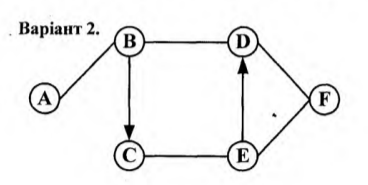
На кожному кроці алгоритм додає потік збільшує шляху до вже наявного потоку. Якщо пропускні спроможності всіх ребер - цілі числа, легко довести по індукції, що і потоки через всі ребра завжди будуть цілими. Отже, на кожному кроці алгоритм збільшує потік принаймні на одиницю, отже, він зійдеться не більше ніж за O (f) кроків, де f - максимальний потік в графі. Можна виконати кожен крок за час O (E), де E - число ребер в графі, тоді загальний час роботи алгоритму обмежено O (E \* f).

Якщо величина пропускної здатності хоча б одного з ребер - ірраціональне число, то алгоритм може працювати нескінченно, навіть не обов'язково сходячись до правильного рішення.

**Завдання:**

Самостійно задати пропускні здатності дуг і побудувати максимальний потік у транспортній мережі.

Знайти мінімальний розріз мережі і перевірити справедливість теореми Форда-Фалкерсона.



#include <iostream>

#include <memory.h>

#include <stdio.h>

using namespace std;

const int MAX\_VERTICES = 40;

int NUM\_VERTICES;

const int INFINITY = 10000;

int f[MAX\_VERTICES][MAX\_VERTICES];

int c[MAX\_VERTICES][MAX\_VERTICES] = {

{ 0, 5, 0, 0, 0, 0},

{ 5, 0, 7, 3, 0, 0},

{ 0, 0, 0, 0, 2, 0},

{ 0, 3, 0, 0, 0, 4},

{ 0, 0, 2, 8, 0, 6},

{ 0, 0, 0, 4, 6, 0}};

int Flow[MAX\_VERTICES];

int Link[MAX\_VERTICES];

int Queue[MAX\_VERTICES];

int QP, QC;

int FindPath(int source, int target)

{

QP = 0; QC = 1; Queue[0] = source;

Link[target] = -1;

int i;

int CurVertex;

memset(Flow, 0, sizeof(int)\*NUM\_VERTICES);

Flow[source] = INFINITY;

while (Link[target] == -1 && QP < QC)

{

CurVertex = Queue[QP];

for (i=0; i<NUM\_VERTICES; i++)

if ((c[CurVertex][i] - f[CurVertex][i])>0 && Flow[i] == 0)

{

Queue[QC] = i; QC++;

Link[i] = CurVertex;

if (c[CurVertex][i]-f[CurVertex][i] < Flow[CurVertex])

Flow[i] = c[CurVertex][i];

else

Flow[i] = Flow[CurVertex];

}

QP++;

}

if (Link[target] == -1) return 0;

CurVertex = target;

while (CurVertex != source)

{

f[Link[CurVertex]][CurVertex] +=Flow[target];

CurVertex = Link[CurVertex];

}

return Flow[target];

}

int MaxFlow(int source, int target)

{

memset(f, 0, sizeof(int)\*MAX\_VERTICES\*MAX\_VERTICES);

int MaxFlow = 0;

int AddFlow;

do

{

AddFlow = FindPath(source, target);

MaxFlow += AddFlow;

} while (AddFlow >0);

return MaxFlow;

}

int MinFlow(int source, int target)

{

memset(f, 0, sizeof(int)\*MAX\_VERTICES\*MAX\_VERTICES);

int MinFlow = 0;

int AddFlow;

do

{

AddFlow = FindPath(source, target);

MinFlow += AddFlow;

} while (AddFlow <0);

return MinFlow;

}

int main()

{

setlocale(0,"");

NUM\_VERTICES = 6;

for(int i=0;i<NUM\_VERTICES;i++){

for(int j=0;j<NUM\_VERTICES;j++){

cout<<"\t["<<c[i][j]<<"]";

}cout<<endl;

}

cout<<endl;

int source, target;

while(1){

cout<<endl;

cout<<"start: "; cin>>source; source--;

cout<<"finish: "; cin>>target; target--;

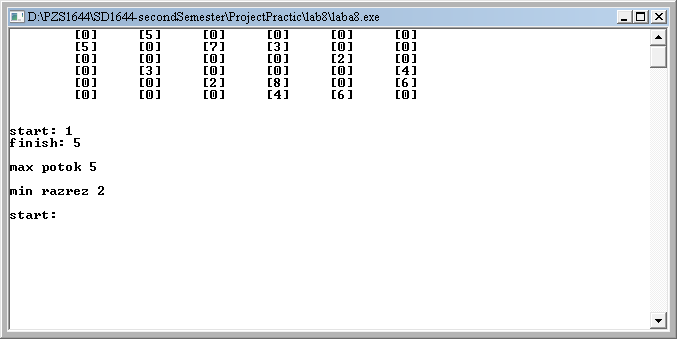
cout<<endl<<"max potok "<<MaxFlow(source, target)<<endl;

cout<<endl<<"min razrez "<<MinFlow(source, target)<<endl;

}

return 0;

}



Висновки: На даній лабораторній роботі ознайомився з методами максимізації потоку даних в обчислювальній системі. Реалізував програму, де був використаний метод Форда-Фалкерсона.