Міністерство освіти, науки, молоді та спорту України

Львівський національний університет імені Івана Франка

Кафедра радіофізики

Звіт

про виконання лабораторної роботи №3

« Пошук найкоротшого шляху між вказаними вершинами за алгоритмом Дейкстри »

Студентки

Групи ФеІ – 22

Литвин Віри

Перевірив

доц. Вельгош С.Р.

Львів 2012

**Тема :** Алгоритм Дейкстри.

**Мета:** Згадати вивчений під час курсу «Дискретна математика» алгоритм пошуку найкоротшого шляху між двома вершинами і написати його програмну реалізацію

**Теоретичний матеріал:**

*Довжиною шляху* в зваженому графі називають суму ваг ребер (дуг), які утворюють цей шлях. Якщо граф не зважений, то вагу кожного ребра (кожної дуги) уважають рівною 1 й отримують раніше введене поняття довжини шляху як кількості ребер (дуг) у ньому.

Задача про найкоротший шлях полягає в знаходженні найкоротшого шляху від заданої початкової вершини ***а*** до заданої вершини ***z***.

Найефективніший алгоритм визначення довжини найкоротшого шляху від фіксованої вершини до будь-якої іншої запропонував 1959 р. датський математик Е. Дейкстра (Е. Dijkstra). Цей алгоритм застосовний лише тоді, коли вага кожного ребра (дуги) додатна.

Нехай G = (V,E) — зважений орієнтований граф, w(vi ,vj) — вага дуги (vi ,vj). Почавши з вершини а, знаходимо віддаль від а до кожної із суміжних із нею вершин. Вибираємо вершину, віддаль від якої до вершини а найменша; нехай це буде вершина v . Далі знаходимо віддалі від вершини а до кожної вершини суміжної з v вздовж шляху, який проходить через вершину v\*. Якщо для якоїсь із таких вершин ця віддаль менша від поточної, то заміняємо нею поточну віддаль. Знову вибираємо вершину, найближчу до а й не вибрану раніше; повторюємо процес.

Описаний процес зручно виконувати за допомогою присвоювання вершинам міток. Є мітки двох типів — тимчасові та постійні. Вершини з постійними мітками групують у множину М, яку називають *множиною позначених вершин*. Решта вершин має тимчасові мітки, і множину таких вершин позначимо як Т, Т= V\M. Позначатимемо мітку (тимчасову чи постійну) вершини v як l(v). Значення постійної мітки l(v) дорівнює довжині найкоротшого шляху від вершини я до вершини v, тимчасової — довжині найкоротшого шляху, який проходить лише через вершини з постійними мітками. Фіксованою початковою вершиною вважаємо вершину а; довжину найкоротшого шляху шукаємо до вершини z.

**Алгоритм Дейкстри**

Наведемо кроки алгоритму.

Крок 1. Присвоювання початкових значень. Виконати *l(a):= 0* та вважати цю мітку постійною. Виконати *l(v):=* ∞ для всіх v ф а й уважати ці мітки тимчасовими. Виконати *х := а, М:= {а}.*

Крок 2. Оновлення міток. Для кожної вершини *v є Т(х)\М* замінити мітки: *l(v):=min{l(v); l(x) + w(x, v)},* тобто оновлювати тимчасові мітки вершин, у які з вершини х іде дуга.

Крок 3. Перетворення мітки в постійну. Серед усіх вершин із тимчасовими мітками знайти вершину з мінімальною міткою, тобто знайти вершину v\* з умови *l(v\*)=min{ l(v)}, v є Т, де Т= V\M*.

Крок 4. Уважати мітку вершини *v* постійною й виконати  (вершину v\* включено в множину М).

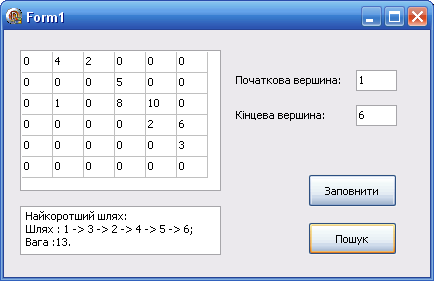
Крок 5. Для пошуку шляху від а до z: якщо x = z, то l(z) — довжина найкоротшого шляху від а до z, зупинитись; якщо x ≠ z, то перейти до кроку 2.

**Практична реалізація:**

Для програмної реалізації я використала середовище Deiphi 7. На формі створила інтерфейс, зображений нище.

Задавати граф для обходу можна одим із методів: матрицею ваг, списком ребер або ж списком суміжності. Ми задаємо матрицю ваг.

Програма виводить шлях, що складається з пронумерованих вершин і довжину цього шляху.



**Висновок**

Алгоритм Дейсктри - найефективніший алгоритм визначення довжини найкоротшого шляху від фіксованої вершини до будь-якої іншої, проте застосовний лише тоді, коли вага кожного ребра (дуги) додатна.

**Додаток. Код програми:**

const

n = 6;

procedure TForm1.Button1Click(Sender: TObject);

var

a:array[1..n,1..n] of longint;

b:array[1..n]of boolean;

d:array[1..n] of longint;

l:array[1..100] of integer;

q, i, j, m, v: integer;

s,p,o:string[50]; k:string;

begin

s:='';

q := StrToIntDef(Edit1.Text, 1);

if (q < 1) or (q > n) then

begin

q := 1;

end;

for i := 1 to n do

for j := 1 to n do

a[j, i] := StrToIntDef(StringGrid1.Cells[i - 1, j - 1], -1);

fillchar(b,sizeof(b),0);

fillchar(d,sizeof(d), 10000);

d[q] := 0;

for i:=1 to n do

begin

m:=1000;

for j:=1 to n do

if ( (d[j] <= m) and (not b[j]) ) then

begin

m:=d[j];

v:=j;

end;

b[v] := true;

for j:=1 to n do

if ((a[v,j]<>0)and(not b[j])and (d[v]+a[v,j]<d[j])) then

begin

d[j]:=d[v]+a[v,j];

end;

s:=s+inttostr(v);

end;

for i := 1 to n do

if ((i=strtoint(Edit2.text)) and (q=strtoint(edit1.text))) then

begin

j:=1;

p:='';

k:=Inttostr(i);

while (strtoint(s[j])<>i) do

begin

p:=p+s[j]+' -> ';

j:=j+1;

end;

p:=p+InttoStr(i);

Memo.Text :=('Найкоротший шлях: ' + #13+#10 +

'Шлях : '+ p + '; ' + #13+#10 +

'Вага :' + IntToStr(d[i]) + '.');

end;

end;

end.