Міністерство освіти, науки, молоді та спорту України

Львівський національний університет імені Івана Франка

Кафедра радіофізики

Звіт

про виконання лабораторної роботи №3

« Методи обходу вершин графа: вглиб та вшир»

Студентки

Групи ФеІ – 22

Литвин Віри

Перевірив

доц. Вельгош С.Р.

Львів 2012

**Тема :** Методи обходу вершин графа. Обхід глиб. Обхід вшир.

**Мета:** вивчити методи швидкого сортування чисел злиттям та Хоара і якнайраціональніше їх практично реалізувати.

**Теоретичний матеріал:**

**Алгори́тм пошуку́ в ширину́**

Якщо задано граф *G* = (*V*, *E*) та початкову вершину *s*, алгоритм пошуку в ширину систематично обходить всі досяжні із *s* вершини. На першому кроці вершина *s* позначається, як пройдена, а в список додаються всі вершини, досяжні з *s* без відвідування проміжних вершин. На кожному наступному кроці всі поточні вершини списку відмічаються, як пройдені, а новий список формується із вершин, котрі є ще не пройденими сусідами поточних вершин списку.

Для реалізації списку вершин найчастіше використовується [черга](http://uk.wikipedia.org/wiki/%D0%A7%D0%B5%D1%80%D0%B3%D0%B0_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85))(динамічна структура даних, що працює за принципом "перший прийшов - перший пішов").

Виконання алгоритму продовжується до досягнення шуканої вершини або до того часу, коли на певному кроці в список не включається жодна вершина. Другий випадок означає, що всі вершини, доступні з початкової, уже відмічені, як пройдені, а шлях до цільової вершини не знайдений.

Алгоритм має назву пошуку в ширину, оскільки «фронт» пошуку (між пройденими та непройденими вершинами) одноманітно розширюється вздовж всієї своєї ширини. Тобто, алгоритм проходить всі вершини на відстані *k* перед тим як пройти вершини на відстані *k*+1.

**Алгори́тм пошуку́ в глибину́** — алгоритм для обходу структури графа.

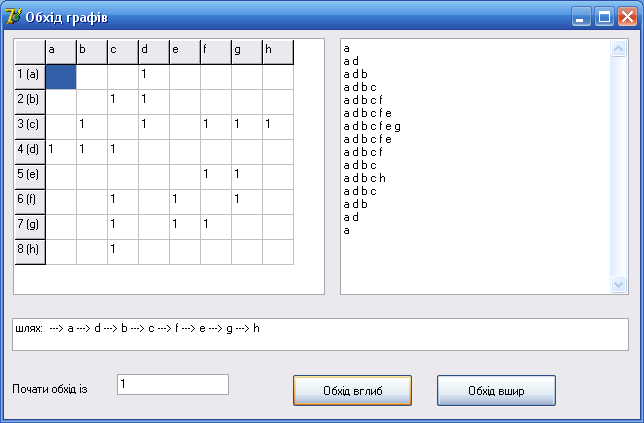
Робота алгоритма починається з кореня дерева (або іншої обраної вершини в графі) і здійснюється обхід в максимально можливу глибину до переходу на наступну вершину.

Для реалізації списку вершин найчастіше використовується [стек](http://uk.wikipedia.org/wiki/%D0%A7%D0%B5%D1%80%D0%B3%D0%B0_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85))(динамічна структура даних, що працює за принципом "останній прийшов - перший пішов").

**Практична реалізація:**

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

Задавати граф для обходу можна матрицею інцидентності.

Програма виводить шлях, що складається з пронумерованих вершин, розділених стрілочкою і вміст структури пам’яті, для обходу вглиб це – стек, а для обходу вшир – черга.

**Висновок**

Обидва методи обходу є досить популярними і мають широку галузь застосування. Через відмінності у реалізації доцільність вибору певного методу залежить від поставленої задачі.

**Додаток. Код програми:**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids, XPMan;

const

n = 8;

mas : array[1..8] of char = ('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h');

type

TForm1 = class(TForm)

Memo1: TMemo;

StringGrid1: TStringGrid;

Edit1: TEdit;

Button1: TButton;

Button2: TButton;

Memo2: TMemo;

XPManifest1: TXPManifest;

Label1: TLabel;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Edit1Exit(Sender: TObject);

private

{ Private declarations }

public

a:array[0..7,0..7] of integer;

b:array of integer;

d:array of integer;

procedure push(x:integer);

procedure pop\_DFS;

procedure pop\_BFS;

function in\_array(x:integer):boolean;

{ Public declarations }

end;

var

Form1: TForm1;

way : String;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

var

i,j:integer;

begin

for i:=1 to n do

StringGrid1.Cells[i,0]:=mas[i];

for i:=1 to n do

StringGrid1.Cells[0,i]:=IntToStr(i)+' (' + mas[i]+ ')';

for i:=0 to n-1 do

for j:=0 to n-1 do

a[i,j]:=0;

StringGrid1.Cells[1,4]:='1';

StringGrid1.Cells[2,3]:='1';

StringGrid1.Cells[4,3]:='1';

StringGrid1.Cells[4,2]:='1';

StringGrid1.Cells[3,6]:='1';

StringGrid1.Cells[3,7]:='1';

StringGrid1.Cells[3,8]:='1';

StringGrid1.Cells[5,6]:='1';

StringGrid1.Cells[5,7]:='1';

StringGrid1.Cells[7,6]:='1';

for i:=1 to n do

for j:=1 to n do

if StringGrid1.Cells[i,j]<>'' then

begin

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

a[j-1,i-1]:=a[i-1,j-1];

StringGrid1.Cells[j,i]:=StringGrid1.Cells[i,j];

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

var

i,j,s:integer;

q:boolean;

str:string;

begin

Memo1.Clear;

Memo2.Clear;

way := '';

q:=False;

SetLength(d,1);

s:=StrToInt(Edit1.Text)-1;

d[0]:=s;

push(s);

Memo1.Lines.Add(mas[b[0]+1]);

repeat

j:=-1;

s:=b[Length(b)-1];

for i:=n-1 downto 0 do

if (a[s,i]=1)and(not in\_array(i)) then j:=i;

if j=-1 then

begin

pop\_DFS;

if Length(b)=0 then

q:=True;

end else

begin

if in\_array(j) then

begin

a[s,j]:=2;

a[j,s]:=2;

end

else

begin

s:=j;

SetLength(d,Length(d)+1);

d[Length(d)-1]:=s;

push(s);

end;

end;

str:='';

if not q then

for i:=0 to Length(b)-1 do

begin

str:=str+ mas[b[i]+1]+' ';

end;

if not q then

Memo1.Lines.Add(str);

until (q);

Memo2.Text := 'шлях: ' + way + ' ';

end;

procedure TForm1.push(x: integer);

begin

SetLength(b,Length(b)+1);

b[Length(b)-1] := x;

way := way + ' ---> ' + mas[x + 1];

end;

procedure TForm1.pop\_DFS;

begin

SetLength(b,Length(b)-1);

end;

function TForm1.in\_array(x: integer): boolean;

var

i,j:integer;

t:boolean;

begin

j:=Length(d)-1;

t:=false;

for i:=0 to j do

if d[i]=x then

begin

t:=true;

end;

in\_array:=t;

end;

procedure TForm1.Button2Click(Sender: TObject);

var

i,j,s:integer;

q:boolean;

str:string;

begin

Memo1.Clear;

Memo2.Clear;

way := '';

q:=False;

SetLength(d,1);

s:=StrToInt(Edit1.Text)-1;

d[0]:=s;

push(s);

Memo1.Lines.Add(mas[b[0]+1]);

repeat

j:=-1;

s:=b[0];

for i:=n-1 downto 0 do

if (a[s,i]=1)and(not in\_array(i)) then j:=i;

if j=-1 then

begin

pop\_BFS;

if Length(b)=0 then

q:=True;

end else

begin

s:=j;

SetLength(d,Length(d)+1);

d[Length(d)-1]:=s;

push(s);

end;

str:='';

if not q then

for i:=0 to Length(b)-1 do

begin

str:=str+mas[b[i]+1]+' ';

end;

if not q then

Memo1.Lines.Add(str);

until (q);

Memo2.Text := 'шлях: ' + way + ' ';

end;

procedure TForm1.pop\_BFS;

var

i,j:integer;

begin

j:=Length(b);

for i:=0 to j-2 do

b[i]:=b[i+1];

SetLength(b,j-1);

end;

procedure TForm1.Edit1Exit(Sender: TObject);

var temp: Integer;

begin

temp := StrToInt(Edit1.Text);

if(temp <= 0) then

Edit1.Text := '1';

end;

end.