**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №4**

з дисципліни  
«Дискретні структури»

на тему «Обхід графа»

Виконав: Перевірила:

студент групи ІП-93 Сергієнко А. А.  
Домінський Валентин Олексійович   
номер залікової книжки: 9311

Київ 2020

**Завдання на лабораторну роботу:**

1. Представити напрямлений граф із заданими параметрами так само, як у

лабораторній роботі №1. Відміна: матриця А за варіантом формується за

командами Scilab:

rand("seed", п1п2п3п4);

T = rand(n,n) + rand(n,n);

A = floor((1.0 - п3\*0.01 - п4\*0.005 - 0.15)\*T)

2. Створити скрипт для Scilab для обходу в глибину при п4 — парному

і для обходу в ширину — при непарному. Обхід починати з вершини, яка має

вихідні дуги. При цьому у скрипті:

— встановити функцію halt у точці призначення номеру черговій

вершині,

— виводити зображення графа у графічному вікні перед кожною

зупинкою по функції halt.

3. Під час обходу графа побудувати дерево обходу. Вивести побудоване

дерево у графічному вікні.

Варіант 9311:  
*п =* 11  
розміщення вершин: колом

**Текст програми для Scilab**

**clf; //clearing canvas**

**clear;**

**TypeOfGraph = 1; //0 is for non-oriented,1 for oriented**

**N1 = 9; //numbers of My Zalikovka to make figure**

**N2 = 3;**

**N3 = 1;**

**N4 = 1;**

**N = 10 + N3; //number of vertexes**

**//N=5**

**//preparing for making canvas**

**//making values for plot2d**

**//x on left is 0**

**ForPlot\_X1 = 0;**

**//x on right is 70**

**ForPlot\_X2 = 70;**

**//y on down is 0**

**ForPlot\_Y1 = 0;**

**//y on upper is 70**

**ForPlot\_Y2 = 70;**

**RadiusOfCircle = 5; //Woah,its radius of circle**

**DiameterOfCircle = 2\*RadiusOfCircle; //Woah,its diameter of circle**

**ColorOfArrow\_1 = 14; //color of arrow\_1 //1 is good**

**ColorOfArrow\_2 = 9; //color of arrow\_2 //3,**

**N1N2N3N4 = N1\*1000 + N2\*100 + N3\*10 + N4**

**//generating matrix for or-graph(from the konspekt)**

**rand("seed", N1N2N3N4);**

**T = rand(N,N) + rand(N,N);**

**A\_oriented = floor((1.0 - N3\*0.01 - N4\*0.005 - 0.15)\*T)**

**//generating matrix for non\_or-graph**

**At=A\_oriented'; //transpose matrix**

**A\_non\_oriented=At+A\_oriented //final non\_oriented matrix(adding regular matrix and transpose)**

**//loop,where we changing 2,which we get by adding**

**//two matrixes,for every element,that = 2**

**for v=1:1:N**

**for v1=1:1:N**

**if A\_non\_oriented(v,v1) == 2**

**A\_non\_oriented(v,v1)=1**

**end**

**end**

**end**

**//choosing matrix and width of arrows**

**if TypeOfGraph == 0**

**Matrix = A\_non\_oriented;**

**WidthOfArrow = 0;**

**elseif TypeOfGraph == 1**

**Matrix = A\_oriented;**

**WidthOfArrow = 30;**

**end**

**//making canvas with our varibles**

**//plot2d([0,200],[0,100],[-1,-1],"022") — формує**

**//прямокутне поле у графічному вікні розміром 100х200 одиниць.**

**//xsetech-ділить вікно на області**

**xsetech([0,0,0.5,1]);**

**plot2d([ForPlot\_X1;ForPlot\_X2],[ForPlot\_Y1;ForPlot\_Y2],0);**

**//looking for a center**

**//(100-0)/2=50**

**X\_O = (ForPlot\_X2-ForPlot\_X1)/2;**

**Y\_O = (ForPlot\_Y2-ForPlot\_Y1)/2;**

**//radius of the big circle,on which we will place our vertexes**

**BigRadius = 2/5\*sqrt((ForPlot\_X2-ForPlot\_X1)\*(ForPlot\_Y2-ForPlot\_Y1))**

**for x=1:1:N//making operations for each element**

**X\_Position = cos(2 \* %pi \* x / N) \* BigRadius + X\_O;//calculating coordinates**

**Y\_Position = sin(2 \* %pi \* x / N) \* BigRadius + Y\_O;**

**//drawing vertexes**

**// xarc(x,y,w,h,a1,a2) — малювання дуги еліпса, який вписаний у прямокутник**

**//з координатами (х,у) верхнього лівого кута і шириною w та висотою h,**

**//яка обмежена променями з кутами а1 і а2, які задаються у 64-х частках градуса.**

**//Наприклад, коло діаметром 4 одиниць з центром у точці (10,20) задається**

**//функцією xarc(8,22,4,4,0,360\*64);**

**xarc(X\_Position,Y\_Position,DiameterOfCircle,DiameterOfCircle,0,360\*64);**

**// xnumb(x,y,data) — вивід числа data починаючи з позиції з координатами**

**//(х,у).**

**xnumb(X\_Position+RadiusOfCircle,Y\_Position-RadiusOfCircle,x);**

**end**

**for i=1:1:N**

**for j=1:1:N**

**if Matrix(i,j) == 1;**

**Position\_X\_I = cos(2 \* %pi \* i / N) \* (BigRadius-RadiusOfCircle) + X\_O;**

**Position\_Y\_I = sin(2 \* %pi \* i / N) \* (BigRadius-RadiusOfCircle) + Y\_O;**

**Position\_X\_J = cos(2 \* %pi \* j / N) \* (BigRadius-RadiusOfCircle) + X\_O;**

**Position\_Y\_J = sin(2 \* %pi \* j / N) \* (BigRadius-RadiusOfCircle) + Y\_O;**

**if i == j**

**Position\_X\_I = cos(2 \* %pi \* i / N) \* (BigRadius) + X\_O;**

**Position\_Y\_I = sin(2 \* %pi \* i / N) \* (BigRadius) + Y\_O;**

**Position\_X\_J = cos(2 \* %pi \* j / N) \* (BigRadius) + X\_O;**

**Position\_Y\_J = sin(2 \* %pi \* j / N) \* (BigRadius) + Y\_O;**

**xarc(Position\_X\_I-RadiusOfCircle,Position\_Y\_I+RadiusOfCircle\*3/4,3/2\*DiameterOfCircle,3/2\*DiameterOfCircle,0,190\*64);**

**//xarrows([x1;x2], [y1;y2], w, c) — стрілка з координатами початку**

**//(x1;y1), кінця (x2;y2), розмірами вістря w і кольором с (с=1 –чорний, 2– синій і**

**//т.д.).**

**xarrows([Position\_X\_I-RadiusOfCircle; Position\_X\_J],[Position\_Y\_I-RadiusOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, 1);**

**else**

**if Matrix(i,j)==Matrix(j,i) && i>j**

**if TypeOfGraph == 1**

**CentrePosition\_X = (Position\_X\_I+RadiusOfCircle+Position\_X\_J+RadiusOfCircle)/2;**

**CentrePosition\_Y = (Position\_Y\_I-RadiusOfCircle+Position\_Y\_J)/2;**

**// xsegs([x1;y1], [x2;y2], c) — пряма з координатами початку [x1;y1],**

**//кінця [x2;y2] і кольором с.**

**if (CentrePosition\_X<X\_O && CentrePosition\_Y<Y\_O)**

**xsegs([Position\_X\_I+RadiusOfCircle; CentrePosition\_X+RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; CentrePosition\_Y+DiameterOfCircle],ColorOfArrow\_2);**

**xarrows([CentrePosition\_X+RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[CentrePosition\_Y+DiameterOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, ColorOfArrow\_2);**

**elseif (CentrePosition\_X>X\_O && CentrePosition\_Y<Y\_O)**

**xsegs([Position\_X\_I+RadiusOfCircle; CentrePosition\_X+RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; CentrePosition\_Y+DiameterOfCircle],ColorOfArrow\_2);**

**xarrows([CentrePosition\_X+RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[CentrePosition\_Y+DiameterOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, ColorOfArrow\_2);**

**elseif (CentrePosition\_X>X\_O && CentrePosition\_Y>Y\_O)**

**xsegs([Position\_X\_I+RadiusOfCircle; CentrePosition\_X-RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; CentrePosition\_Y-DiameterOfCircle],ColorOfArrow\_2);**

**xarrows([CentrePosition\_X-RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[CentrePosition\_Y-DiameterOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, ColorOfArrow\_2);**

**elseif (CentrePosition\_X<X\_O && CentrePosition\_Y>Y\_O)**

**xsegs([Position\_X\_I+RadiusOfCircle; CentrePosition\_X+RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; CentrePosition\_Y-DiameterOfCircle],ColorOfArrow\_2);**

**xarrows([CentrePosition\_X+RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[CentrePosition\_Y-DiameterOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, ColorOfArrow\_2);**

**end**

**else**

**xarrows([Position\_X\_I+RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, ColorOfArrow\_1);**

**end**

**elseif TypeOfGraph == 0**

**else**

**xarrows([Position\_X\_I+RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, ColorOfArrow\_1);**

**end**

**end**

**end**

**end**

**end**

**//Start=1**

**function [OneOfTheForest,MatrixOfVertexes] = bfs(A\_oriented, Past,Start)**

**OneOfTheForest=zeros(N,N);**

**MatrixOfVertexes=zeros(N,N);**

**Top=1**

**Now=1**

**Back=0**

**Head=0**

**Back=Back+1;**

**Turn(Back)=Start;**

**Past(Start)=1;**

**while Back~=Head**

**Head=Head+1;**

**Start=Turn(Head);**

**for i=1:1:N**

**if Past(i)~=1 && A\_oriented(Start,i)==1**

**Back=Back+1;**

**Turn(Back)=i;**

**Past(i)=1;**

**Now=Back;**

**end**

**end**

**for j=Top+1:1:Now**

**OneOfTheForest(Start,Turn(j))=1;**

**end**

**Top=Now;**

**end**

**for j=1:1:N**

**MatrixOfVertexes(j,Turn(j))=1;**

**end**

**endfunction**

**for i=1:1:N**

**Past(i)=0;**

**end**

**[OneOfTheForest,Number\_Matrix]=bfs(A\_oriented, Past, 1);**

**disp(OneOfTheForest);**

**disp(Number\_Matrix);**

**xsetech([0.5,0,0.5,1]);**

**plot2d([ForPlot\_X1;ForPlot\_X2],[ForPlot\_Y1;ForPlot\_Y2],0);**

**//looking for a center**

**//(100-0)/2=50**

**X\_O = (ForPlot\_X2-ForPlot\_X1)/2;**

**Y\_O = (ForPlot\_Y2-ForPlot\_Y1)/2;**

**//radius of the big circle,on which we will place our vertexes**

**BigRadius = 2/5\*sqrt((ForPlot\_X2-ForPlot\_X1)\*(ForPlot\_Y2-ForPlot\_Y1))**

**for x=1:1:N//making operations for each element**

**X\_Position = cos(2 \* %pi \* x / N) \* BigRadius + X\_O;//calculating coordinates**

**Y\_Position = sin(2 \* %pi \* x / N) \* BigRadius + Y\_O;**

**//drawing vertexes**

**// xarc(x,y,w,h,a1,a2) — малювання дуги еліпса, який вписаний у прямокутник**

**//з координатами (х,у) верхнього лівого кута і шириною w та висотою h,**

**//яка обмежена променями з кутами а1 і а2, які задаються у 64-х частках градуса.**

**//Наприклад, коло діаметром 4 одиниць з центром у точці (10,20) задається**

**//функцією xarc(8,22,4,4,0,360\*64);**

**xarc(X\_Position,Y\_Position,DiameterOfCircle,DiameterOfCircle,0,360\*64);**

**// xnumb(x,y,data) — вивід числа data починаючи з позиції з координатами**

**//(х,у).**

**xnumb(X\_Position+RadiusOfCircle,Y\_Position-RadiusOfCircle,x);**

**end**

**CountOfLines=1;**

**for i=1:1:N**

**for j=1:1:N**

**if OneOfTheForest(i,j) == 1;**

**Position\_X\_I = cos(2 \* %pi \* i / N) \* (BigRadius-RadiusOfCircle) + X\_O;**

**Position\_Y\_I = sin(2 \* %pi \* i / N) \* (BigRadius-RadiusOfCircle) + Y\_O;**

**Position\_X\_J = cos(2 \* %pi \* j / N) \* (BigRadius-RadiusOfCircle) + X\_O;**

**Position\_Y\_J = sin(2 \* %pi \* j / N) \* (BigRadius-RadiusOfCircle) + Y\_O;**

**if OneOfTheForest(i,j)==OneOfTheForest(j,i) && i>j**

**else**

**xarrows([Position\_X\_I+RadiusOfCircle; Position\_X\_J+RadiusOfCircle],[Position\_Y\_I-RadiusOfCircle; Position\_Y\_J-RadiusOfCircle], WidthOfArrow, CountOfLines+10);**

**FirstCoord=((Position\_X\_I+RadiusOfCircle)+(Position\_X\_J+RadiusOfCircle))/2**

**SecondCoord=((Position\_Y\_I-RadiusOfCircle)+(Position\_Y\_J-RadiusOfCircle))/2**

**disp(FirstCoord)**

**disp(SecondCoord)**

**//if CountOfLines > 2 && CountOfLines < 9**

**// xnumb(FirstCoord,SecondCoord,CountOfLines+1)**

**//elseif CountOfLines == 9**

**// xnumb(FirstCoord,SecondCoord,3)**

**//elseif CountOfLines <= 2**

**// xnumb(FirstCoord,SecondCoord,CountOfLines)**

**//else**

**xnumb(FirstCoord,SecondCoord,CountOfLines)**

**//end**

**CountOfLines=CountOfLines+1**

**halt('Press a key')**

**end**

**end**

**end**

**end**

**Згенерована матриця суміжності**

1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0.

0. 0. 1. 1. 1. 0. 1. 1. 0. 1. 1.

0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 0.

0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1.

0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0.

0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0.

0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0.

1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.

1. 0. 1. 0. 1. 1. 0. 0. 1. 0. 1.

0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0.

**Матриця дерева обходу:**

0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0.

0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1.

0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.

0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

**Матриця відповідності вершин і одержаної нумерації:**

1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.

0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.

0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.

0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.

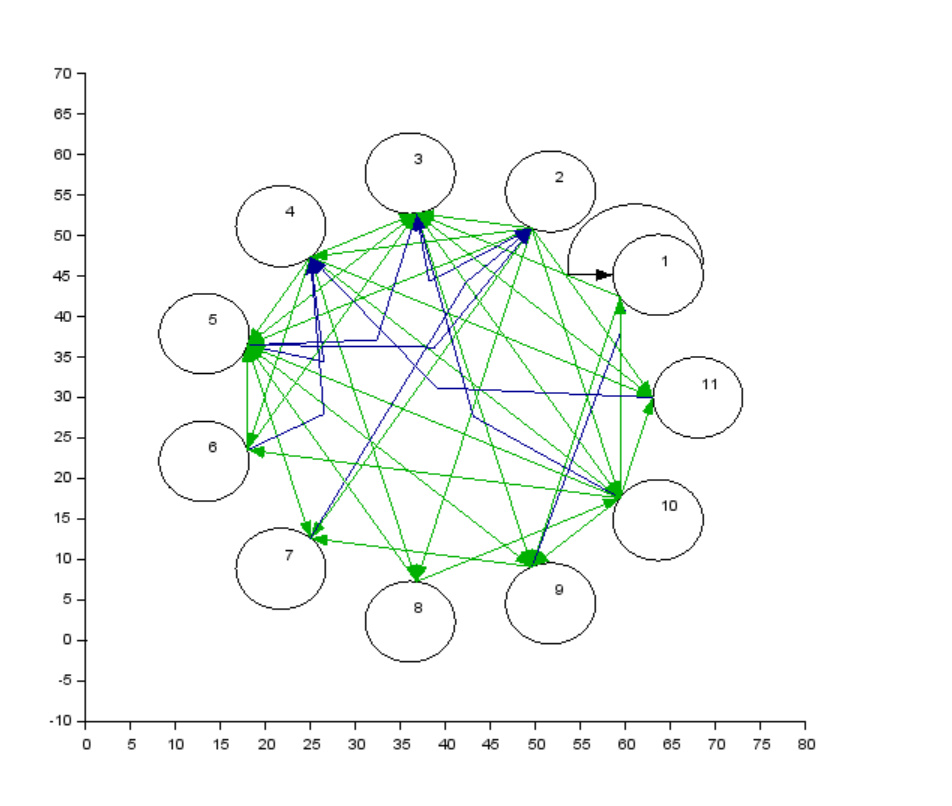
0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.

0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.

**Результати виконання програми:**

Орграф:



**Дерево обходу:**

