**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 №1

з дисципліни  
«Дискретні структури»

на тему «Графічне представлення графів»

Виконав: Перевірила:

студент групи ІП-95 Сергієнко А. А.

Грибинюк Олександр Сергійович  
номер залікової книжки: 9504

Київ 2020

**Завдання на лабораторну роботу**

1. Представити у програмі напрямлений і ненапрямлений графи з заданими параметрами:

— число вершин *n*;

— розміщення вершин;

— матриця суміжності A.

2. Створити скрипт для формування зображення напрямленого і ненапрямленого графів у графічному вікні Scilab.

Варіант 9504:  
*п = 14*  
розміщення вершин: трикутником при n = 4

**Текст програми для Scilab**

*//Генерування матриці напрямленого графу*

rand("seed", 9504);

T = rand(10,10) + rand(10,10);

A = floor((1.0-0.02-4\*0.005-0.25)\*T);

disp(A);

*//Матриця суміжності ненапрямленого графу*

for i=1:10

for j=1:10

if(A(i,j)==1 && A(j,i)~=1) then

A(j,i)=1;

end

end

end

disp(A);

clf;

plot2d([0,100], [0,100],0);

*//Функціх створення кола навколо точки і вписування номеру вершини*

function circle(**x**, **y**, **r**)

xarc(**x**-**r**, **y**+**r**, 2\***r**, 2\***r**, 0, 64\*360);

endfunction;

function **coord**=vertex(**x**, **y**, **n**)

global Vmatrix;

circle(**x**,**y**,4);

xnumb(**x**-1,**y**-1,**n**);

**coord** = [**x**; **y**];

Vmatrix(:, **n**) = **coord**;

endfunction;

*//Задання центрів кіл та номерів вершин*

vertex(10,10, 1);

vertex(30,10, 2);

vertex(50,10, 3);

vertex(70,10, 4);

vertex(90,10, 5);

vertex(25,30, 10);

vertex(35,45, 9);

vertex(50,65, 8);

vertex(75,30, 6);

vertex(65,45, 7);

*//Функція з’єднання точок(вказуємо лише центри кіл і чи будуть вони мати напрям)*

function connect(**x1**, **y1**, **x2**, **y2**, **direct**)

if (**direct** == 0) then

xr = **x2** - **x1**

yr = **y2** - **y1**

k = yr / xr

xx = sqrt((4^2)/(1+k^2));

yy = xx \* k;

**x1** = **x1**+xx;

**y1**=yy+**y1**;

if ((**y2** == **y1**) && (**x2** < **x1**)) then

xsegs([**x1**-2\*xx;**x2**+xx],[**y2**;**y1**]);

elseif ((**x2** < **x1**) && (**y2** < **y1**))

xsegs([**x1**-xx;**x2**+xx],[**y1**-2\*yy;**y2**+yy]);

elseif ((**x2** < **x1**) && (**y2** > **y1**))

xsegs([**x1**-2\*xx;**x2**+xx],[**y1**-2\*yy;**y2**+yy]);

elseif (**x1** ~= **x2**) xsegs([**x1**;**x2**-xx],[**y1**;**y2**-yy]);

else xsegs([**x1**;**x2**],[**y1**-4;**y2**+4]);

end

end

if (**direct** == 1)then

xr = **x2** - **x1**

yr = **y2** - **y1**

k = yr / xr

xx = sqrt((4^2)/(1+k^2));

yy = xx \* k;

**x1** = **x1**+xx;

**y1**=yy+**y1**;

if ((**y2** == **y1**) && (**x2** < **x1**)) then

xarrows([**x1**-2\*xx;**x2**+xx],[**y2**;**y1**],50,0);

elseif ((**x2** < **x1**) && (**y2** < **y1**))

xarrows([**x1**-xx;**x2**+xx],[**y1**-2\*yy;**y2**+yy],50,0);

elseif ((**x2** < **x1**) && (**y2** > **y1**))

xarrows([**x1**-2\*xx;**x2**+xx],[**y1**-2\*yy;**y2**+yy],50,0);

elseif (**x1** ~= **x2**) xarrows([**x1**;**x2**-xx],[**y1**;**y2**-yy],50,0);

else xarrows([**x1**;**x2**],[**y1**-4;**y2**+4],50,0);

end

end

endfunction;

*// Зв’язок графу самим з собою*

function back(**x**, **y**, **r**)

if (**x**<50) then

circle(**x**-6,**y**,**r**-2);

elseif (**x**>50) circle(**x**+6,**y**,**r**-2);

end

endfunction

function connection(**x1**, **y1**, **x2**, **y2**)

xsegs([**x1**;**x2**+15],[**y1**-4;**y2**]);

xarrows([**x2**+15;**x2**+4],[**y2**;**y2**], 50, 0);

endfunction

*//Функція оберненого зв’язку*

function backward(**x1**, **y1**, **x2**, **y2**)

xr = **x2** - **x1**

yr = **y2** - **y1**

k = yr / xr

xx = sqrt((4^2)/(1+k^2));

yy = xx \* k;

**x1** = **x1**+xx;

**y1** = yy+**y1**;

xz = (**x1**+**x2**)/2;

yz = (**y1**+**y2**)/2;

xsegs([**x1**-2\*xx;45],[**y1**-2\*yy;yz]);

xarrows([45;**x2**+xx],[yz;**y2**+yy],50,0);

endfunction;

*//Застосовуємо функцію з’єднання (в залежності від останнбого параметра «0» чи «1» отримаємо ненапрямлений і напрямлений граф відповідно)*

connect(10,10,65,45,1);

connect(30,10,50,10,1);

connect(50,10,35,45,1);

connect(35,45,90,10,1);

connect(35,45,70,10,1);

connect(70,10,75,30,1);

connect(25,30,90,10,1);

connect(65,45,35,45,1);

connect(65,45,50,65,1);

connect(70,10,90,10,1);

connect(75,30,65,45,1);

connect(50,65,50,10,1);

connect(10,10,25,30,1);

connection(50,65, 25, 30);

backward(70,10,35,45);

back(35,45,4);

back(65,45,4);

back(90,10,4);

**Згенеровані матриці суміжності**

Матриця суміжності напрямленого графу:

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

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

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

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

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

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

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

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

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

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

Матриця суміжності ненапрямленого графу:

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

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

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

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

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

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

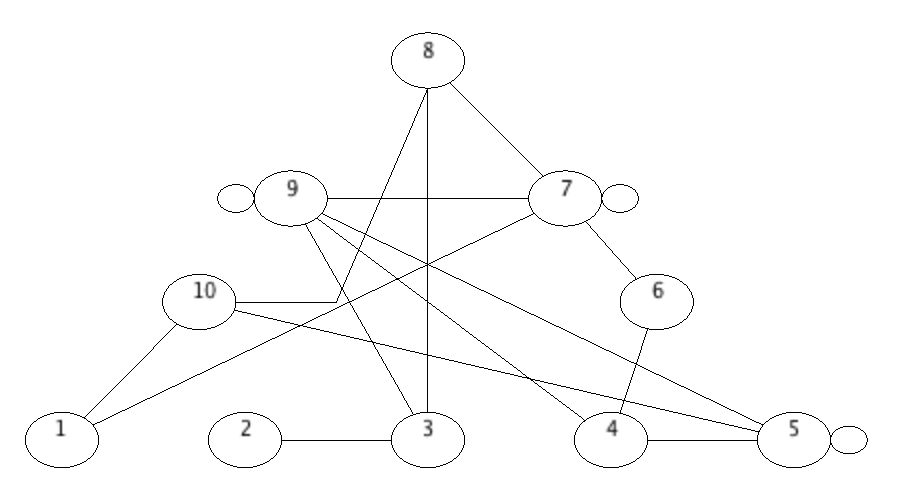
1. 0. 0. 0. 0. 1. 1. 0. 1. 0.

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

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

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

**Результати виконання програми**

Ненапрямлений граф:

Напрямлений граф: