**Ministerul Educației al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

## RAPORT

**la disciplina: “Matematica discretă”**

**Lucrare de laborator Nr. 6**

**Tema: “Drum Hamiltonian”**

**A verificat : Dohotaru Leonid**

**A efectuat: st. Brăduleac Vadim , gr. AI-151**

**Chișinău 2016**

Se numeşte graf, ansamblul format dintr-o mulţime finită *X* şi o aplicaţie *F* a lui *X* în *X*. Se notează *G = (X,F)*. Numărul elementelor mulţimii *X* determină ordinul grafului finit. Dacă *card X = n*, graful *G = (X,F)* se numeşte *graf finit de ordinul n*. Elementele mulţimii *X* se numesc *vârfurile* grafului. Geometric, vârfurile unui graf le reprezentăm prin puncte sau cerculeţe. Perechea de vârfuri *(x,y)* se numeşte *arc*; vârful *x* se numeşte originea sau extremitatea iniţială a arcului *(x,y)*, iar vârful *y* se numeşte extremitatea finală sau terminală. Un arc *(x,y)* îl reprezentăm geometric printr-o săgeată orientată de la vârful *x* la vârful *y*.

Dacă un vârf nu este extremitatea nici unui arc el se numeşte *vârf izolat*, iar dacă este extremitatea a mai mult de două arce - *nod*. Un arc pentru care extremitatea iniţială coincide cu cea finală se numeşte *buclă*.

Arcele unui graf le mai notăm şi cu *u1*, *u2*,..., iar mulţimea arcelor grafului o notăm cu *U*. Se observă că mulţimea *U* a tuturor arcelor unui graf determină complet aplicaţia *F*, precum şi reciproc, aplicaţia *F* determină mulţimea *U* a arcelor grafului. Un graf *G* poate fi dat fie prin ansamblul *(X,F)* fie prin ansamblul *(X,U)*.

Două *arce* se zic *adiacente* dacă sunt distincte şi au o extremitate comună. Două *vârfuri* se zic *adiacente* dacă sunt distincte şi sunt unite printr-un arc.

Un arc *(x,y)* se spune că este *incident* cu vârful *x* *spre exterior* şi este *incident* cu vârful y *spre interior*.

Fie *G = (X,F)* şi *x ∈ X*. Mulţimea tuturor arcelor incidente cu *x* spre exterior (interior) se numeşte semigradul exterior (interior) a lui *x* şi se notează *d+x (d\_x)*. Dacă pentru un vârf *x*, *d+x=0* sau *d\_x=0* atunci el se numeşte vârf terminal

*x1*

*x4*

*x2*

*x3*

*x5*

*x6*

*x7*

*u1*

*u2*

*u3*

*u4*

*u5*

*u6*

*u7*

*u8*

Fig. 4.1. Exemplu de graf

Într-un graf *G = (X,U)* se numeşte drum un şir de arce *(u1,...,uk)*, astfel încât extremitatea terminală a fiecărui arc *uI* coincide cu extremitatea iniţială a arcului următor *ui+1*. Un drum care foloseşte o singură dată fiecare arc al său se numeşte drum simplu. Un drum care trece o singură dată prin fiecare vârf al său se numeşte drum elementar. Lungimea unui drum este numărul de arce din care este compus drumul.

Un drum elementar ce trece prin toate vârfurile grafului se numeşte drum hamiltonian. Un drum simplu ce conţine toate arcele grafului se numeşte drum eulerian. Un drum finit pentru care vârful iniţial coincide cu vârful terminal se numeşte circuit.

Graful obţinut din graful iniţial suprimând cel puţin un vârf al acestuia precum şi toate arcele incidente cu el se numeşte subgraf. Graful obţinut suprimând cel puţin un arc se numeşte graf parţial.

Un graf se numeşte complet dacă oricare ar fi *x* şi *y* din *X* există un arc de la *x* la *y* sau de la *y* la *x*.

Un graf *G = (X,F)* se zice tare conex dacă pentru orice *x*, *y* *∈ X* (*x* diferit de *y*) există un drum de la *x* la *y* sau că oricare pereche de vârfuri *x, y* cu *x* diferit de *y* se află pe un circuit.

Fie *G = (X,F)* şi *x ∈ X*. Mulţimea *Cx* formată din toate vârfurile *xi* *∈ X* pentru care există un circuit ce trece prin *x* şi *xi* se numeşte componentă tare conexă a lui *G* corespunzătoare vârfului *x*.

Componentele tari conexe ale unui graf *G = (X,F)* constituie o partiţie a lui *X*.

Noţiunile introduse sunt valabile pentru grafurile orientate.

În cazul în care orientarea arcelor nu are nici o importanţă graful se va numi *neorientat*. Arcele se vor numi *muchii*, drumul - *lanţ*, circuitul - *ciclu*. La fel se introduce noţiunea de lanţ elementar şi simplu, lanţ Euler şi hamiltonian. Graful tare conex se va numi conex.

Cele două concepte de graf orientat şi graf neorientat se pot sprijini în practică unul pe altul. De la un graf orientat se poate trece la omologul său neorientat când se abordează o problemă ce nu presupune orientarea şi invers, dacă se precizează orientarea.

Unui graf orientat simetric i se poate asocia un graf neorientat, legătura dintre două vârfuri *x* şi *y* realizată de cele două arce *(x,y)* şi *(y,x)* de sensuri contrarii înlocuindu-se cu muchia *[x,y]*. De asemenea un graf neorientat poate fi identificat cu mai multe grafuri orientate înlocuind fiecare muchie cu două arce orientate în sens opus.

Fie *G = (X,U)* un graf neorientat şi *x ∈ X*. Se numeşte gradul vârfului *x* numărul muchiilor care au o extremitate în vârful *x*. Se notează *g(x)*. Un vârf este izolat dacă *g(x) = 0*.

**Listingul programului:**

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main()

{ int \*v[10],\*a,\*p,t1; int i,j,n,t[10];

char c; int ch[10][10]={0};

int i1; system("color 9");

printf("DATI NUMARUL DE VIRFURI\n");

scanf("%i",&n);

for(i=0;i<n;i++)

v[i]=(int\*)malloc((n-1)\*sizeof(int));

printf("INTRODUCETI LISTA DE ADIACENTA\n");

for(i=0;i<n;i++)

{printf("V %i->: ",i+1);

a=v[i];j=1;

while (j!=0){ scanf("%i",&j);

\*(v[i]++)=j; } v[i]=a;}

for(i=0;i<n;i++) {a=v[i];

while (\*v[i]!=0)

{ ch[i][\*v[i]-1]=1; v[i]++;} v[i]=a;}

for(i=0;i<n;i++)

free(v[i]);

for(i=0;i<n;i++)

{ for(j=0;j<n;j++)

t[j]=0;

t1=0;

while (t1==0){t1=1;

for(j=0;j<n;j++)

if ((ch[i][j]==1)&&(t[j]==0)) { t1=0;

for(i1=0;i1<n;i1++)

if ((ch[i][i1]==0)&&(ch[j][i1])==1)

ch[i][i1]=1;

t[j]=1;break;

}} }

for(i=0;i<n;i++)

t[i]=0;

printf("MATRICEA DRUMURILOR ESTE:\n");

for(i=0;i<n;i++)

{ for(j=0;j<n;j++){ if (ch[i][j]==1)

t[i]++;

printf("\t%i",ch[i][j]); }

printf("\n"); }

i1=0;

for(i=0;i<n;i++)

i1+=t[i];

if (i1==((n-1)\*n)/2)

{ printf("DRUMUL HAMILTONIAN ESTE:\n");

for(i=0;i<n;i++) { i1=t[0];

for(j=0;j<n;j++)

if (t[j]>i1)

i1=t[j];

for(j=0;j<n;j++)

if (t[j]==i1) {

printf("V%i -> ",j+1); t[j]=-1; break; } }

printf("\b\b ");}

else printf("NU EXITA DRUM HAMILTONIAN PENTRU GRAFUL DAT\n");

getch(); }

**Concluzie :** Efecuand aceasta lucrare de laborator,am obtinut practica de lucrucu metodele de reprezentare ale grafului,si metodele lui de stocare in memoria calculatorului.

Pe parcursul lucrarii, am ajuns la concluzia ca cea mai simpla metoda de introducere a grafului de la tastatura este listade adiacenta,de asemenea aceasta metoda utilizeaza si cel mai putin din memoria calculatorului.Dupa parerea mea, cel mai usor putem desena graful,daca ne orientam dupa matricea de incidenta,insa ea ocupa un volum mai mare din memoria calculatorului.