**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**



**Лабораторна робота №6**

з дисципліни

«Дискретна математика»

**Виконав:**

студент групи КН-110

Король Орест

**Викладач:**

Мочурад Л.І.

Львів – 2017 рік

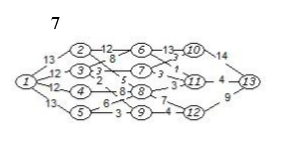
# *Лабораторна робота 6*

*****Тема:***** Потоки в мережах

*Мета роботи:* набуття практичних вмінь та навичок з пошуку максимального потоку в мережі.

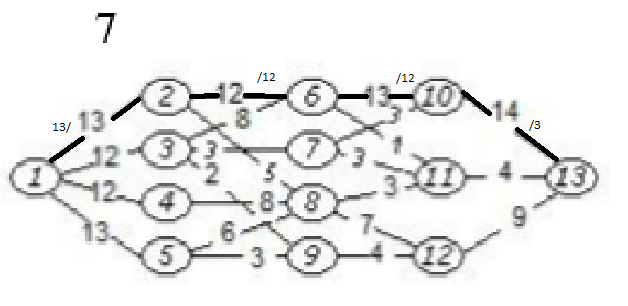
**Варіант №7**

Завдання № 1. Побудувати повний потік, а потім скорегувати його до найбільшого (дуги спрямовані зліва направо).

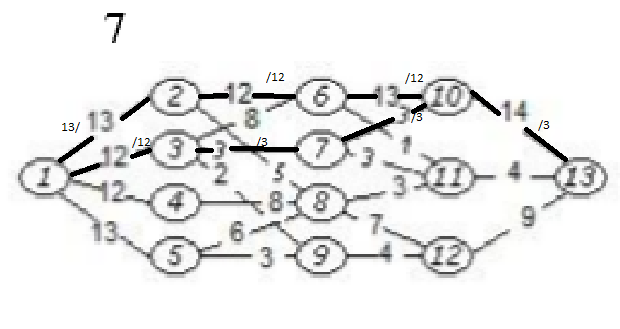


Розв’язання:

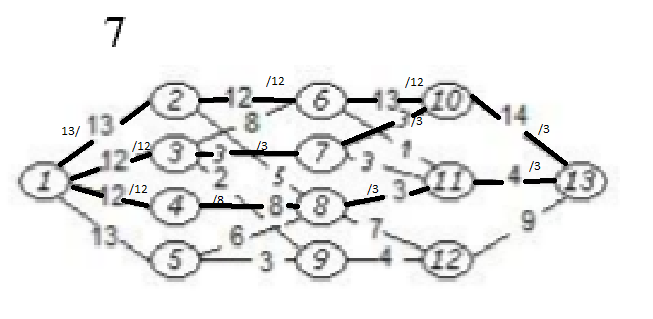
Крок 1:



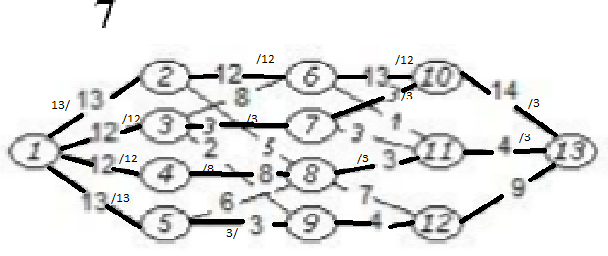
Крок 2:

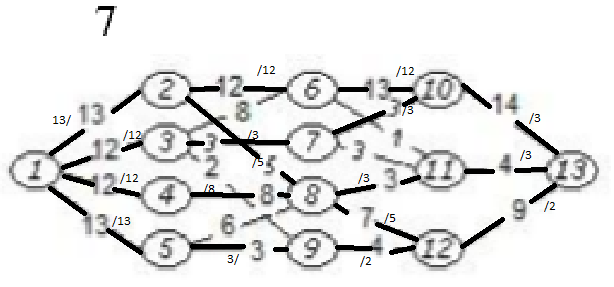


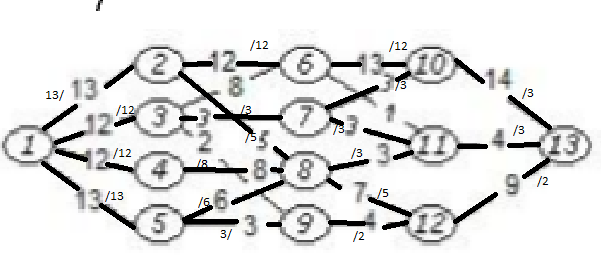
Крок 3:

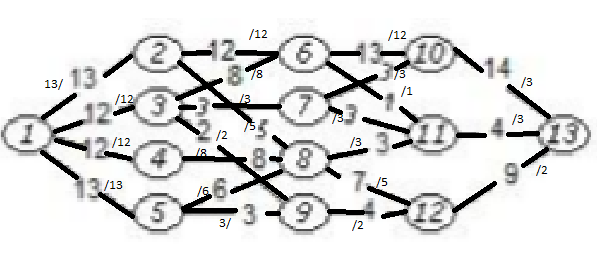


Крок 4:

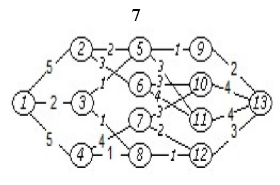
Крок 5:

Крок 6:

Крок 7(додаєм решту шляхів)

V = 11

Завдання №2. Написати програму, яка знаходить максимальний потік в мережі. Протестувати за розробленою програмою мережу згідно свого варіанту.



**Код програми:**

#include <iostream>

#include <limits.h>

#include <string.h>

#include <queue>

using namespace std;

bool bfs(int rGraph[13][13], int s, int t, int parent[])

{

bool visited[13];

memset(visited, 0, sizeof(visited));

queue <int> q;

q.push(s);

visited[s] = true;

parent[s] = -1;

while (!q.empty())

{

int u = q.front();

q.pop();

for (int v=0; v<13; v++)

{

if (visited[v]==false && rGraph[u][v] > 0)

{

q.push(v);

parent[v] = u;

visited[v] = true;

}

}

}

return (visited[t] == true);

}

int fordFulkerson(int graph[13][13], int s, int t)

{

int u, v;

int rGraph[13][13];

for (u = 0; u < 13; u++)

for (v = 0; v < 13; v++)

rGraph[u][v] = graph[u][v];

int parent[13];

int max\_flow = 0;

while (bfs(rGraph, s, t, parent))

{

int path\_flow = INT\_MAX;

for (v=t; v!=s; v=parent[v])

{

u = parent[v];

path\_flow = min(path\_flow, rGraph[u][v]);

}

cout << "Parent movement height : ";

cout << "13";

for (v=t; v != s; v=parent[v])

{

u = parent[v];

rGraph[u][v] -= path\_flow;

rGraph[v][u] += path\_flow;

cout <<"-"<<parent[v] + 1;

}

max\_flow += path\_flow;

cout<< endl <<"Min path length : "<< max\_flow <<endl;

}

return max\_flow;

}

int main()

{

int graph[13][13] = {

{0,5,2,5,0,0,0,0,0,0,0,0,0},

{5,0,0,0,2,3,0,0,0,0,0,0,0},

{2,0,0,0,1,0,0,1,0,0,0,0,0},

{5,0,0,0,0,0,4,1,0,0,0,0,0},

{0,2,1,0,0,0,0,0,1,0,3,0,0},

{0,3,0,0,0,0,0,0,0,3,4,0,0},

{0,0,0,4,0,0,0,0,0,3,0,2,0},

{0,0,1,1,0,0,0,0,0,0,0,1,0},

{0,0,0,0,1,0,0,0,0,0,0,0,2},

{0,0,0,0,0,3,3,0,0,0,0,0,4},

{0,0,0,0,3,4,0,0,0,0,0,0,4},

{0,0,0,0,0,0,2,1,0,0,0,0,3},

{0,0,0,0,0,0,0,0,2,4,4,3,0}};

cout << endl << "Max path lenght : " << fordFulkerson(graph, 0, 12)<<endl;

return 0;

}

**Bисновок:** я навчився знаходити потоки в мережах та оперувати ними.