**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

Факультет комп’ютерних наук

Кафедра «Інтелектуальних інформаційних систем»

**ЗВІТ**

*з лабораторної роботи № 9*

Дисципліна "Теорія алгоритмів"

Напрям підготовки: 122 Комп’ютерні науки

ЛР.ПЗ.09-201.1910103

***Cтудент*** *\_\_\_Грабовський Є.О.*

*(підпис)*

*\_\_\_\_\_\_\_\_13.12.2020\_\_\_\_\_\_\_\_ \_\_*

*(дата)*

***Викладач*** *\_\_\_Гожий О.П.*

*(підпис)*

*(дата)*

**Практична робота №9**

**Тема: Пошук максимального потоку у мережі.**

**Мета:**

1. Опанувати особливості конкретного середовища програмування.

2. Опанувати методику створення програмного продукту.

3. Реалізувати алгоритм пошуку максимального потоку у мережі.



Рис. 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вар. №** | **Ваги ребер**  **Пропускна спроможність ребер** | | | | | | | | | | | | | | | | | | | **Маршрут** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **А – B** |
| 8 | 18  2 | 18  2 | 4  3 | 21  20 | 9  4 | 9  4 | 8  3 | 9  8 | 6  6 | 9  8 | 13  7 | 18  2 | 9  4 | 14  3 | 8  3 | 9  8 | 9  6 | 5  2 | 9  6 | 1-10 |

Для знаходження максимального потоку було обрано алгоритм Форда-Фалкерсона.

**Лістинг коду:**

using System;

using System.Collections.Generic;

public class MaxFlow

{

static readonly int V = 11;

bool Bfs(int[,] rGraph, int s, int t, int[] parent)

{

var visited = new bool[V];

for (var i = 0; i < V; ++i)

visited[i] = false;

var queue = new List<int> {s};

visited[s] = true;

parent[s] = -1;

while (queue.Count != 0)

{

var u = queue[0];

queue.RemoveAt(0);

for (var v = 0; v < V; v++)

{

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

{

queue.Add(v);

parent[v] = u;

visited[v] = true;

}

}

}

return (visited[t]);

}

int FordFulkerson(int[,] graph, int s, int t)

{

s--;

t--;

int u, v;

var rGraph = new int[V, V];

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

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

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

var parent = new int[V];

var maxFlow = 0;

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

{

var pathFlow = int.MaxValue;

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

{

u = parent[v];

pathFlow = Math.Min(pathFlow, rGraph[u, v]);

}

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

{

u = parent[v];

rGraph[u, v] -= pathFlow;

rGraph[v, u] += pathFlow;

}

maxFlow += pathFlow;

}

return maxFlow;

}

public static void Main()

{

int[,] graph = {

{ 0, 18, 4,21, 0, 0, 0, 0, 0, 0, 0},

{18, 0, 9, 0, 0, 18, 0, 0, 0, 0, 0},

{ 4, 9, 0, 9, 13, 8,18, 6, 0, 0, 0},

{21, 0, 9, 0, 9, 0, 0, 0, 9, 0, 0},

{ 0, 0, 13, 9, 0, 0, 9, 0, 0, 5, 0},

{ 0, 18, 8, 0, 0, 0, 0, 9, 0, 0, 0},

{ 0, 0, 18, 0, 9, 0, 0, 14, 0, 0, 0},

{ 0, 0, 6, 0, 0, 9,14, 0, 0, 0, 8},

{ 0, 0, 0, 9, 0, 0, 0, 0, 0, 9, 0},

{ 0, 0, 0, 0, 5, 0, 0, 0, 9, 0, 0},

{ 0, 0, 0, 0, 0, 0, 0, 8, 0, 9, 0}

};

var m = new MaxFlow();

Console.WriteLine("Максимальный поток сети: " +

m.FordFulkerson(graph, 1, 10));

Console.ReadKey();

}

}

**Результат виконання:**



**Складність алгоритму Форда-Фалкерсона:**

O(|E|f), де E — кількість ребер у графі, f — максимальний потік у графі,

**Висновок:**

В ході проведення лабораторної роботи № 9 було досліджено знаходження максимального потоку в графі, скористалися алгоритмом Форда-Фалкерсона для вирішення цієї задачі, знайшли скадність алгоритму.