**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Лабораторна робота №4

з курсу «Дискретні моделі в САПР»:

**ПОТОКОВІ АЛГОРИТМИ**


**Виконав:**

Ст.гр.КН-409

Погуляєв В.В.


.

Львів – 2023

## Мета роботи

Метою даної лабораторної роботи є вивчення потокових алгоритмів.

## Теоретичні відомості

Останнім часом значно зросла зацікавленість учених та практиків потоковими моделями. Це пов’язано із впровадженням та активним розвитком різноманітних територіально розподілених систем: трубопровідних, транспортних, телекомунікаційних та ін. Основою таких систем є певна мережа (мережа трубопроводів, доріг, каналів зв’язку тощо), в якій циркулюють певні потоки (потоки речовин, транспорту, даних тощо), тому задачі, які доводиться розв’язувати при проектуванні та експлуатації систем з мережною структурою, часто зводяться до розробки математичних моделей розподілу потоків та постановки і розв’язання відповідних оптимізаційних задач. Відомі моделі розподілу потоків у мережах базуються на поняттях теорії графів. Це пов’язано з тим, що граф дає можливість наочно відобразити структуру мережі, а параметри його вузлів і дуг – представити основними числовими характеристиками її елементів. Набір характеристик залежить від природи модельованої системи, а також характеру розв’язуваних задач, однак у потокових моделях їх, як правило, представляють такими параметрами, як зовнішній потік у вузлі, потік по дузі, пропускна здатність дуги, вартість передавання одиниці потоку по дузі тощо. Потокові задачі, як правило, зводяться до пошуку такого розподілу потоків у мережі, при якому б забезпечувався екстремум деякого критерію. При цьому мають враховуватися обмеження, що накладаються умовами збереження потоків у вузлах і неперевищення потоками пропускної здатності дуг. Типовими потоковими задачами є задача про потік мінімальної вартості, про максимальний потік, транспортна задача, задача про призначення та інші. Для їх розв’язання розроблено чимало ефективних алгоритмів, сформувався навіть відповідний напрям обчислювальних методів під назвою потокового програмування.

## Лабораторне завдання

Реалізувати алгоритм Форда – Фалкерсона.

## Код програми

Метод пошуку шляху:

```
public int findPath(int[][] cap, boolean[] vis, int u, int t, int f){
    if(u == t){
        return f;
    }
    vis[u] = true;
    for (int v = 0; v < vis.length; v++){
        if(!vis[v] && cap[u][v] > 0){
            int df = findPath(cap, vis, v, t, Math.min(f, cap[u][v]));
            if(df > 0){
                cap[u][v] -= df;
                cap[v][u] += df;
                return df;
```

```
            }
        }
    }
    return 0;
}
```

Посилання на GitHub – https://github.com/flipppflopp/DM_Pohuliaiev

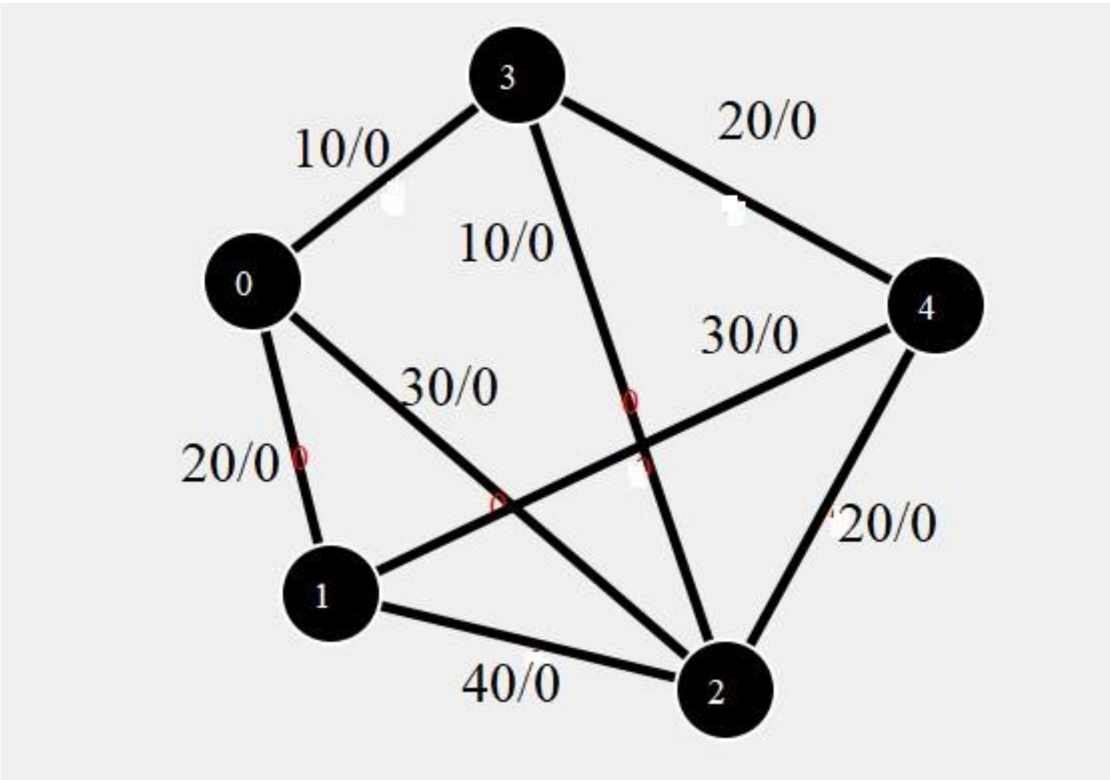
**Аналіз результатів**

**Аналітичний розв’язок:**

Матриця:

```
0 20 30 10 0
0 0 40 0 30
0 0 0 10 20
0 0 0 0 20
0 0 0 0 0
```

Граф:




Аналітичний обрахунок:

Необхідно знайти максимальний потік з вершини 1 до вершини 5.

Іттерація №1:

Починаємо з вершини 1. Нам необхідно вибрати шлях значення якого є найбільшим серед шляхів, які йдуть від вершини 1,отже переходимо до вершини 3 і ставим їй позначку (30,1). З вершини 3 прямуємо до вершини 5, отже ставимо позначку вершині 5(20,3). Ми дійшли до кінцевої точки, випишемо наші значення шляхів: f = {30, 20}.

Наш шляху: 1->3->5.

Тепер необхідно знайти $f_{min}$. $f_{min} = \min\{30,20\} = 20$

Тепер необхідно зменшити значення ребер, які складають наскрізний шлях в напрямку потоку, і збільшити значення шляху, які розміщення в протилежному напрямку.

Отже, в нас виходить така матриця:

0 20 10 10 0

0 0 40 0 30

20 0 0 10 0

0 0 0 0 20

0 0 20 0 0

Іттерація №2:

Виконуючи аналогічні кроки, ми получаємо такий шлях:

1->2->3->4->5

$f_{min} = \min\{20,40,10,20\} = 10$.

Оновлена матриця:

0 10 10 10 0

10 0 30 0 30

20 10 0 0 0

0 0 10 0 10

0 0 20 10 0

Іттерація №3:

Шлях:

1->2->5.

$f_{min} = \min\{10,30\} = 10$.

Оновлена матриця:

0 0 10 10 0

0 0 30 0 20

20 10 0 0 0

0 0 10 0 10

0 10 20 10 0

Іттерація №4:

Шлях:

1->3->2->5.

$f_{min}$ = min{10, 10, 20} = 10.

Оновлена матриця:

0 0 0 10 0

0 0 40 0 10

30 0 0 0 0

0 0 10 0 10

0 20 20 10 0

Іттерація №5:

Шлях:

1->4->5.

$f_{min}$ = min{10, 10} = 10.

Оновлена матриця:

0 0 0 0 0

0 0 40 0 10

30 0 0 0 0

10 0 10 0 0

0 20 20 20 0

Іттерація №6:

Так як всі значення шляхів від точки 1 рівні 0, то необхідно обчислити максимальний потік знайдених наскрізних шляхів:

$F = f_1 + f_2 + \ldots + f_n$ .

F = 20 + 10 + 10 + 10 + 10 = 60 – максимальний об’єм потоку в мережі.

Результат виконання програми для аналітичного обрахунку(Від вершини 1 до вершини 5):

```
Maximum flow from source 0 to sink 4 is 60

Process finished with exit code 0
```

Результат виконання програми використовуючи тестовий файл I4_1.txt(Від вершини 1 до вершини 8):

```
C:\Users\nazar\.jdks\openjdk-18.0.1\bin\java.e
Maximum flow from source 0 to sink 7 is 55

Process finished with exit code 0
```

Результат виконання програми використовуючи тестовий файл I4_2.txt(Від вершини 1 до вершини 8):

```
C:\Users\nazar\.jdks\openjdk-18.0.1\bin\java.ex
Maximum flow from source 0 to sink 7 is 70

Process finished with exit code 0
```

**Висновок**

В ході виконання лабораторної роботи, вивчив потокові алгоритми, реалізував програмно алгоритм Форда – Фалкерсона.