

| НУЛП, ІКНІ, САПР | | Тема | оцінка | підпис |
|-------------------------|---|--------------------|--|--------|
| КН-414 | 3 | Потокові алгоритми | | |
| Юнусов Ю. М. | | | | |
| № залікової: 16083062 | | | | |
| Дискретні моделі в САПР | | | Викладач: к.т.н., асистент Кривий Р.З. | |

Мета:

Вивчення поточкових алгоритмів.

Завдання: Написати програму для демонстрації роботи обраного поточкового алгоритму.

Варіант 13. Алгоритм Форда-Фалкерсона.

Теоретичні відомості:

Алгоритм Форда-Фалкерсона є одним з способів рішення задачі побудови максимального потоку в мережі.

Опис:

Знайти будь-який шлях, що збільшується. Збільшити потік по всіх його ребрах на мінімальну з їх залишкових пропускних здатностей. Повторювати, поки є шлях, що збільшується. Алгоритм працює тільки для цілих пропускних здатностей. В іншому випадку, він може працювати нескінченно довго, не сходячись до правильної відповіді.

Складність:

Залежить від алгоритму пошуку шляху, що збільшується. Потребує $O(\max |f|)$ таких пошуків.

Для пошуку шляху, що збільшується я використав алгоритм пошуку в ширину.

Алгоритм має таке практичне значення: рішення транспортної задачі, наприклад, потрібно перевезти з початкової вершини мережі в кінцеву вантаж по дугах мережі за мінімальний час, при цьому по кожній дузі не можна перевозити вантажу більше фіксованого об'єму.

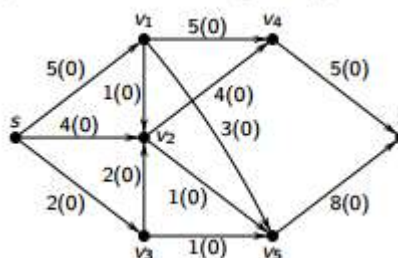


Рис.1 Початковий граф

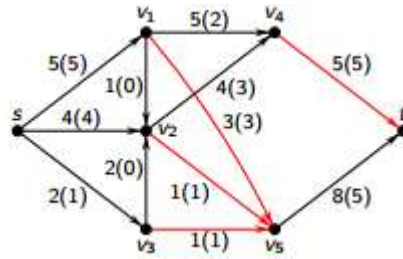


Рис.2 Граф після роботи алгоритму (макс. потік 10)

Робота з програмою:

Для запуску роботи алгоритму потрібно натиснути кнопку 'Старт'.

Для вводу всіх точок записаних у програму зразу використовується кнопка 'Додати всі'.

Для очищення списку ребер та графу потрібно натиснути 'Очистити'.

Для додавання ребер по одному вводимо на панелі додавання відповідні дані (дві вершини та вагу ребра) та нажимаємо 'Додати'.

На панелі задання початку та кінця задаємо номери початкової та кінцевої вершин.

У верхньому тексті боксі відображається результат роботи алгоритму, а у нижньому виводиться інформація про роботу програми.

Справа на елементі Канвас відображається побудований граф.

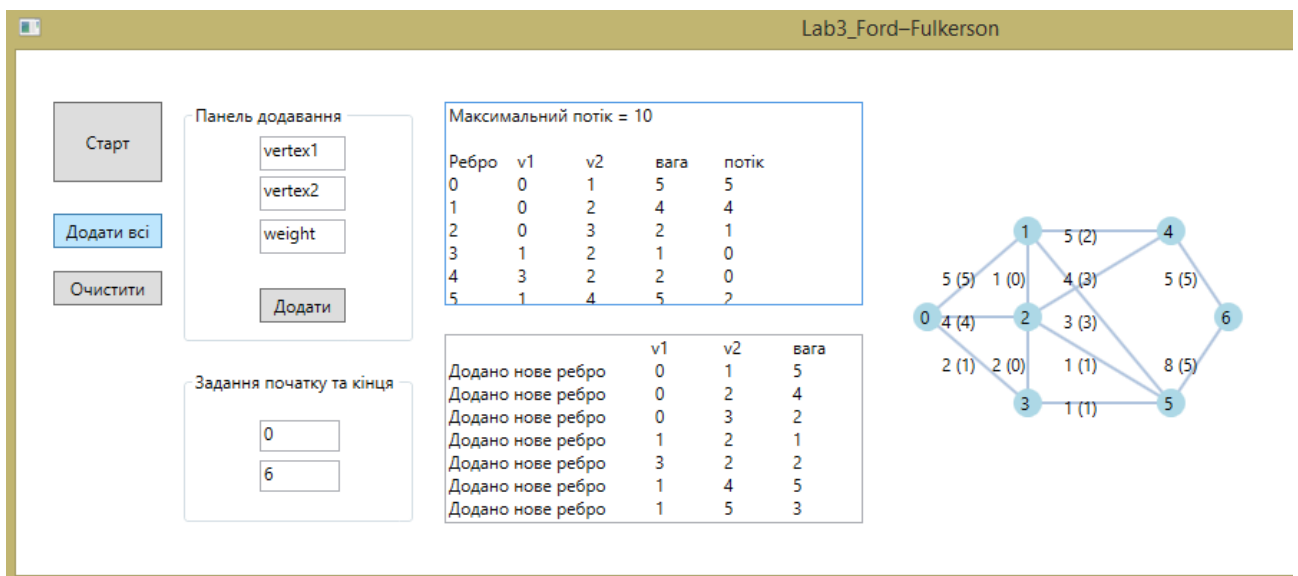


Рис.3 Результат роботи програми

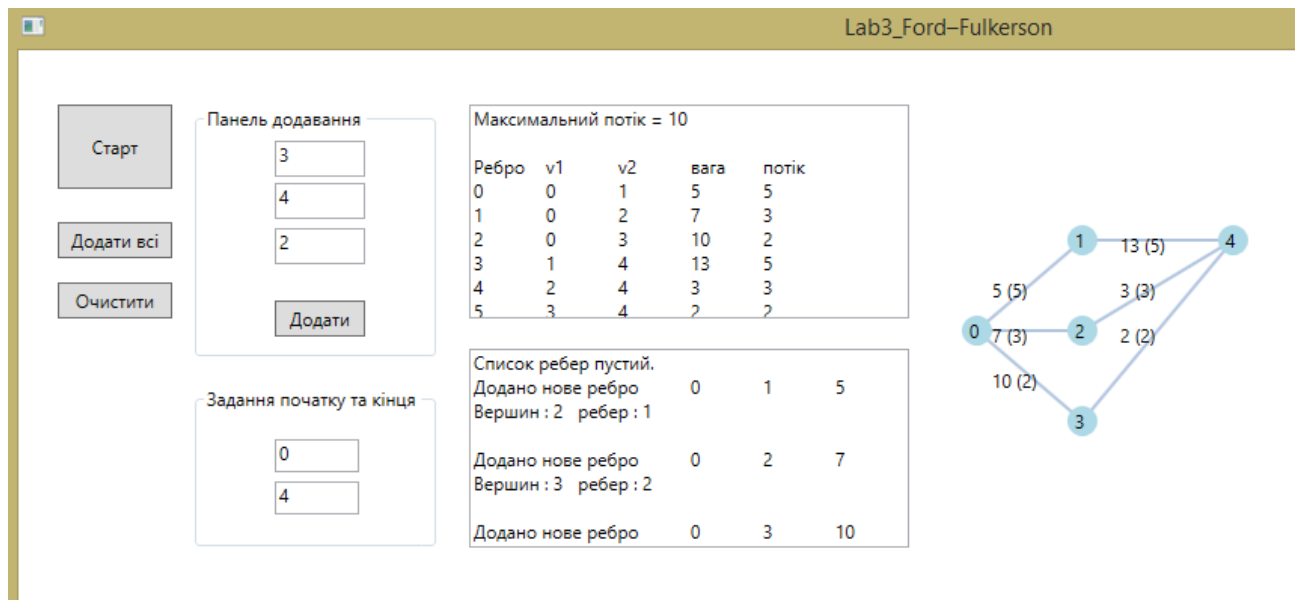


Рис.4 Результат роботи програми для інших даних

Фрагмент програми:

```
int FordFulkerson(int source, int sink) {
    int i, j, u;
    int max_flow = 0;
    for (i = 0; i < nodes; i++) {
        for (j = 0; j < nodes; j++) {
            flow[i, j] = 0;
        }
    }
    while (bfs(source, sink)) {
        int increment = 10000;
        for (u = nodes - 1; pred[u] >= 0; u = pred[u]) {
            increment = min(increment, capacity[pred[u], u] - flow[pred[u], u]);
        }
        for (u = nodes - 1; pred[u] >= 0; u = pred[u]) {
            flow[pred[u], u] += increment;
            flow[u, pred[u]] -= increment;
        }
        max_flow += increment;
    }
    return max_flow;
}
```

Висновок: На цій лабораторній роботі було здійснено ознайомлення з потоковим алгоритмом Форда-Фалкерсона та програмно реалізовано його.