

GRAFLARDA FLOYD-WARSHALL ALGORITMINING AHAMIYATI

Farmonov Sherzodbek Raxmonjonovich

*Farg'ona davlat universiteti amaliy matematika va
informatika kafedrası katta o'qituvchisi*

farmonovsh@gmail.com

Mirzaabdullayeva Kamolatxon Abdulxafiz qizi

Farg'ona davlat universiteti 2-kurs talabasi

mirzaabdullayevakamolatxon@gmail.com

Annotatsiya: Floyd-Warshall algoritmi — bu graf nazariyasida, ayniqsa, yo'l masofalarini hisoblashda ishlatiladigan klassik algoritmdir. U, barcha juft nuqtalar orasidagi har bir tugun orasidagi eng qisqa yo'lni topish uchun mo'ljallangan bo'lib, grafdagi har bir tugun orasidagi eng qisqa masofani aniqlashda yordam beradi. Algoritmning asosiy g'oyasi, har bir juft tugun orasidagi masofani iteratsiya orqali yangilash va eng qisqa yo'llarni topishdan iborat. Bu algoritmning asosiy afzalligi shundaki, u barcha juft nuqtalarni orasidagi eng qisqa yo'llarni topish uchun yagona ishvo berish amaliyoti biln bajariladi.

Kalit so'zlar: Floyd-Warshall algoritmi. Graf nazariyasi. Kislorodning qisqa yo'li. Yig'ishdagi eng qisqa yo'llar. Grafning barcha juft nuqtalari orasidagi eng qisqa yo'l. Dinamik dasturlash. Algoritmik komplekslik. C# dasturlash tili. Yo'l masofasini matritsalar yordamida hisoblash. Grafni ifodalash (matritsa yoki ro'yxat). Floyd-Warshallning implementatsiyasi kodini yozish. Grafning mustahkamligi. Eng qisqa yo'l masofalarini topish. Asosiy algoritmik g'oyalar. Floyd-Warshall algoritmining arxitekturası. Algoritmni optimallashtirish. Dasturdagi murakkablik darajalari. Floyd-Warshall algoritmining amaliy qo'llanilishi.

Аннотация: Алгоритм Флойда-Уоршелла — это классический алгоритм, используемый в теории графов, особенно для вычисления расстояний между путями. Он предназначен для нахождения кратчайшего пути между каждой парой узлов в графе, помогая определить кратчайшее расстояние между любыми двумя узлами графа. Основная идея алгоритма заключается в том, чтобы обновлять расстояния между каждой парой узлов через итерации и находить кратчайшие пути. Основное преимущество этого алгоритма заключается в том, что он выполняет одну операцию для нахождения кратчайших путей между всеми парами узлов.

Ключевые слова: Алгоритм Флойда-Уоршелла. Теория графов. Кратчайший путь кислорода. Кратчайшие пути в сборке. Кратчайший путь между всеми парами узлов графа. Динамическое программирование. Алгоритмическая сложность. Язык программирования C#. Вычисление расстояния между путями с помощью матриц. Представление графа (матрица или список). Реализация алгоритма Флойда-Уоршелла. Устойчивость графа. Нахождение кратчайших расстояний. Основные алгоритмические идеи. Архитектура алгоритма Флойда-Уоршелла. Оптимизация алгоритма. Уровни сложности программы. Практическое применение алгоритма Флойда-Уоршелла.

Abstract: The Floyd-Warshall algorithm is a classical algorithm used in graph theory, particularly for calculating path distances. It is designed to find the shortest path between every pair of vertices, helping to determine the shortest distance between all pairs of nodes in a graph. The core idea of the algorithm is to iteratively update the distance between each pair of vertices and find the shortest paths. The main advantage of this algorithm is that it performs a single operation to find the shortest paths between all pairs of vertices.

Keywords: Floyd-Warshall algorithm. Graph theory. Shortest path of oxygen. Shortest paths in a collection. Shortest path between all pairs of nodes in a graph. Dynamic programming. Algorithmic complexity. C# programming language. Computing path distances using matrices. Graph representation (matrix or list). Implementation of the Floyd-Warshall algorithm. Graph stability. Finding the shortest path distances. Core algorithmic ideas. Architecture of the Floyd-Warshall algorithm. Algorithm optimization. Complexity levels of the program. Practical applications of the Floyd-Warshall algorithm.

Kirish: Floyd-Warshall algoritmi graf nazariyasida eng qisqa yo'llarni hisoblash uchun ishlatiladigan klassik algoritmdir. U barcha juft tugunlar orasidagi eng qisqa yo'llarni topishga imkon beradi va bu algoritmi juda samarali bo'lib, ko'plab real dunyo masalalarida qo'llaniladi. Masalan, yo'l tarmog'ida trafikni boshqarish, kompyuter tarmoqlaridagi eng qisqa yo'llarni hisoblash va boshqalar. Ushbu maqolada biz Floyd-Warshall algoritmining nazariy asoslarini ko'rib chiqamiz va uni C# dasturlash tilida qanday amalga oshirishni ko'rsatamiz.

Floyd-Warshall Algoritmi Nima?

Floyd-Warshall algoritmi — bu **dinamik dasturlash** (dynamic programming) usulida ishlaydigan, grafdagi barcha juft tugunlar orasidagi eng qisqa yo'llarni topish uchun ishlatiladigan algoritmdir. Algoritmi barcha tugunlar orasidagi eng qisqa yo'llarni topish uchun bir vaqtning o'zida barcha mumkin bo'lgan yo'llarni hisoblashni amalga oshiradi.

Algoritmning o'ziga xos xususiyati shundaki, u **qisqa yo'llarni iterativ ravishda** yangilab boradi, bunda har bir tugun bir necha bor qilingan hisob-kitoblarda eng qisqa yo'lni tanlash uchun tekshiriladi.

Algoritmning Ishlash Prinsipi

Floyd-Warshall algoritmi, oddiy qilib aytganda, har bir juft tugunlar orasidagi eng qisqa yo'lni hisoblash uchun uchta asosiy qadamni bajaradi:

1. **Boshlang'ich holat:** Algoritm, grafdagi barcha tugunlar orasidagi masofalarni saqlovchi matritsa (masofa matritsasi) yaratadi. Agar ikki tugun orasida bevosita bog'lanish bo'lsa, masofa to'g'ridan-to'g'ri bog'lanish qiymati bo'ladi, aks holda u cheksiz qiymatga teng bo'ladi.
2. **Iteratsiya:** Har bir tugunni qadam-baqadam tekshirish orqali, u orqali o'tgan eng qisqa yo'llarni hisoblash. Har bir yangi yo'lni hisoblashda, o'sha tugun orqali o'tadigan yo'lni yangilash.
3. **Natija:** Natijada, grafdagi barcha juft tugunlar orasidagi eng qisqa yo'llar matritsasi olinadi.

Algoritmning Murakkabligi

Floyd-Warshall algoritmining murakkabligi $O(n^3)$ bo'lib, bunda n — grafdagi tugunlar sonidir. Algoritmning bu murakkabligi uni katta graflar uchun sekinlashtiradi, ammo kichik yoki o'rta hajmdagi graflarda samarali ishlaydi.

Floyd-Warshall Algoritmini C# Tilida Amaliyotda Qo'llash

Endi Floyd-Warshall algoritmini C# dasturlash tilida qanday amalga oshirishni ko'rib chiqamiz. Quyidagi kod misolida biz 4 ta tugundan iborat grafni ko'rib chiqamiz va har bir juft tugunlar orasidagi eng qisqa yo'llarni hisoblaymiz.

C# Tilida Floyd-Warshall Algoritmi

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        // Tugunlar soni
```

```
        int V = 4;
```

```
        // Masofalar matritsasini yaratish
```

```
        int[,] dist = new int[V, V];
```

```
        // Matritsani boshlang'ich qiymatlarda to'ldirish
```

```
        for (int i = 0; i < V; i++)
```

```
{  
    for (int j = 0; j < V; j++)  
    {  
        if (i == j)  
            dist[i, j] = 0; // O'z-o'zidan masofa 0 bo'ladi  
        else  
            dist[i, j] = int.MaxValue; // Bog'lanmagan tugunlar uchun cheksiz  
            qiymat  
    }  
}
```

// Graflar orasidagi bog'lanishlarni qo'shish

dist[0, 1] = 3; // 0 dan 1 ga

dist[0, 2] = 5; // 0 dan 2 ga

dist[1, 2] = 1; // 1 dan 2 ga

dist[1, 3] = 7; // 1 dan 3 ga

dist[2, 3] = 2; // 2 dan 3 ga

// Floyd-Warshall algoritmini ishlatish

FloydWarshall(dist, V);

// Natijalarni chiqarish

Console.WriteLine("Eng qisqa yo'llar matritsasi:");

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

{

for (int j = 0; j < V; j++)

{

if (dist[i, j] == int.MaxValue)

Console.Write("INF ".PadRight(6));

else

Console.Write(dist[i, j].ToString().PadRight(6));

}

Console.WriteLine();

}

}

// Floyd-Warshall algoritmi

static void FloydWarshall(int[,] dist, int V)

```
{
    for (int k = 0; k < V; k++) // O'rtadagi tugunni ko'rib chiqamiz
    {
        for (int i = 0; i < V; i++) // Har bir boshlang'ich tugun uchun
        {
            for (int j = 0; j < V; j++) // Har bir tugun orasidagi masofani yangilash
            {
                if (dist[i, k] != int.MaxValue && dist[k, j] != int.MaxValue &&
dist[i, j] > dist[i, k] + dist[k, j])
                {
                    dist[i, j] = dist[i, k] + dist[k, j]; // Yangilangan eng qisqa yo'l
                }
            }
        }
    }
}
```

Kodning Izohli Tushuntirishlari:

1. **Boshlang'ich matritsa:** dist matritsasi, grafdagi barcha tugunlar orasidagi masofalarni saqlaydi. O'zaro bog'lanmagan tugunlar orasidagi masofa int.MaxValue (cheksiz) deb belgilangan.
2. **Grafni boshlang'ich holatda to'ldirish:** Ma'lum bog'lanishlar o'rnatilgan (masalan, 0 dan 1 ga 3, 1 dan 2 ga 1 va hokazo).
3. **Floyd-Warshall algoritmi:** 3 o'lchovli sikl yordamida har bir juft tugunlar orasidagi eng qisqa masofalar hisoblanadi. Agar biror tugun orqali o'qish orqali masofa qisqaradigan bo'lsa, matritsa yangilanadi.
4. **Natijalarni chiqarish:** Algoritm natijasida barcha juft tugunlar orasidagi eng qisqa yo'llar ekranga chiqariladi.

Kutilgan Natija:

Agar yuqoridagi kodni bajarilsa, quyidagi matritsa ekranga chiqariladi:

Eng qisqa yo'llar matritsasi:

0	3	4	6
INF	0	1	3
INF	INF	0	2
INF	INF	INF	0

Bu matritsada:

- 0 dan 2 gacha bo'lgan eng qisqa yo'l masofasi 4,
- 1 dan 3 gacha bo'lgan eng qisqa yo'l masofasi 3 ekanligini ko'rishimiz mumkin.

Floyd-Warshall Algoritmining Afzalliklari va Kamchiliklari**Afzalliklari:**

- Har bir juft tugunlar orasidagi eng qisqa yo'llarni bir martada topadi.
- Graflarda cheksiz bog'lanishlar yoki o'zaro yo'llar bo'lsa ham samarali ishlaydi.

Kamchiliklari:

- Algoritmning vaqt murakkabligi $O(n^3)$ bo'lgani uchun katta hajmdagi graflarda ishlashda samaradorligi kamayadi.
- Har bir juft tugun orasidagi yo'lni hisoblashda vaqt sarfini ko'paytiradi.

Floyd-Warshall algoritmi graf nazariyasida juda foydali vosita bo'lib, barcha juft tugunlar orasidagi eng qisqa yo'llarni hisoblashga imkon beradi. C# dasturlash tilida ushbu algoritmni amalga oshirish oson va samarali bo'lib, uni ko'plab amaliy masalalarda qo'llash mumkin. Algoritmning afzalliklari va kamchiliklarini inobatga olgan holda, uni kerakli hol.

Foydalanilgan adabiyotlar:

1. Marcin Jamro. C# Data Structures and Algorithms. Second Edition. Published by Packt Publishing Ltd., in Birmingham, UK. 2024. – 349 p.
2. Дж.Эриксон. Алгоритмы.: – М.: " ДМК Пресс ", 2023. – 528 с.
3. Hemant Jain. Data Structures & Algorithms using Kotlin. Second Edition. in India. 2022. – 572 p.
4. Н. А. Тюкачев, В. Г. Хлебостроев. C#. Алгоритмы и структуры данных: учебное пособие для СПО. – СПб.: Лань, 2021. – 232 с.
5. Mykel J. Kochenderfer. Tim A. Wheeler. Algorithms for Optimization. Published by The MIT Press., in London, England. 2019. – 500 p.
6. Рафгарден Тим. Совершенный алгоритм. Графовые алгоритмы и структуры данных. – СПб.: Питер, 2019. - 256 с.
7. Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э. Структуры данных и алгоритмы. – М.: Вильямс, 2018. – 400 с.
8. Дж.Хайнеман, Г.Поллис, С.Стэнли. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО "Альфа-книга", 2017. — 432 с.
9. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. В CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (Т. 2, Выпуск 12, сс. 71–74). Zenodo.
10. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. Theoretical aspects in the formation of pedagogical sciences, 2(22), 90-96
11. Raxmonjonovich, F. S. (2023). Array ma'lumotlar tizimini talabalarga o'qitishda Blockchain metodidan foydalanish. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 541-547.

12. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 425-429.
13. Raxmonjonovich, F. S. (2023). Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 434-438.
14. Raxmonjonovich, F. S. (2023). Dasturlash tillarida fayllar bilan ishlash mavzusini Blended Learning metodi yordamida o'qitish. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 464-469.
15. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 475-481.
16. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 482-486.
17. Raxmonjonovich, F. S., & Ravshanbek o'g'li, A. A. (2023). Zamonaviy dasturlash tillarining qiyosiy tahlili. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 430-433.
18. Raxmonjonovich, F. S. (2023). C# dasturlash tilida fayl operatsiyalari qo'llashning qulayliklari haqida. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 439-446.
19. Raxmonjonovich, F. S. (2023). C# tilida ArrayList bilan ishlashning afzalliklari. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 470-474.
20. Farmonov Sherzodbek Raxmonjonovich, & Rustamova Humoraxon Sultonbek qizi. (2024). C# DASTURLASH TILIDA TO'PLAMLAR BILAN ISHLASH. Ta'lim Innovatsiyasi Va Integratsiyasi, 11(10), 210–214. Retrieved from <http://web-journal.ru/index.php/ilmiy/article/view/2480>.
21. Raxmonjonovich, F. S., & Ravshanbek o'g'li, A. A. (2023). Zamonaviy dasturlash tillarining qiyosiy tahlili. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 430-433.
22. Farmonov, S., & Rasuljonova, Z. (2024). OB'EKTGA YO'NALTIRILGAN DASTURLASH ZAMONAVIY DASTURLASHNING ASOSI SIFATIDA. Центральноеазиатский журнал образования и инноваций, 3(1), 83-86.
23. Farmonov, S., & Ro'zimatov, J. (2024). DASTURLASH TILLARINI O'RGANISHDA ONLINE TA'LIM PLATFORMALARIDAN FOYDALANISH. Theoretical aspects in the formation of pedagogical sciences, 3(1), 5-10.