



INTERNET TARMOG'IDA ENG QISQA YO'LNI TOPISHDA
FLOYD-UORSHAL ALGORITMIDAN FOYDALANISH

Sh.R.Farmonov

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

farmonovsh@gmail.com

J.U.Ismoilov

Farg'ona davlat universiteti talabasi

javohir20060612@gmail.com

Anotatsiya: *Floyd-Uorshal algoritmi, graf nazariyasi sohasida eng qisqa yo'llarni aniqlash uchun ishlatiladigan samarali va keng tarqalgan metodlardan biridir. Mazkur maqola Floyd-Uorshal algoritmining asosiy printsiplari, uning vaqt va xotira murakkabligi, dasturlashda qo'llanishi, shuningdek, amaliyotdagi turli sohalarda, jumladan, transport, tarmoq tizimlari, ijtimoiy tarmoqlar va ilmiy tadqiqotlarda qo'llanilishi haqida batafsil ma'lumot beradi.*

Kalit so'zlar: *Floyd-Uorshal algoritmi, eng qisqa yo'l, graf nazariyasi, transport tizimi, tarmoq tahlili, ijtimoiy tarmoq, ilmiy tadqiqotlar, murakkablik, algoritmik yondashuvlar, yo'l tarmog'i, marshrutizatsiya, dasturlash, optimallashtirish, ko'p agentli tizimlar, real hayotdagi qo'llanish.*

Abstract: *The Floyd-Warshall algorithm is one of the most efficient and widely used methods for finding the shortest paths in graph theory. This article provides detailed information on the core principles of the Floyd-Warshall algorithm, its time and memory complexity, its application in programming, and its use in various real-world domains, including transportation, network systems, social networks, and scientific research.*

Keywords: *Floyd-Warshall algorithm, shortest path, graph theory, transportation system, network analysis, social network, scientific research, complexity, algorithmic approaches, road network, routing, programming, optimization, multi-agent systems, real-life applications.*

Аннотация: *Алгоритм Флойда-Уоршелла является одним из самых эффективных и широко используемых методов для нахождения кратчайших путей в теории графов. В данной статье представлена подробная информация о ключевых принципах алгоритма Флойда-Уоршелла, его временной и пространственной сложности, его применении в программировании, а также его использовании в различных реальных областях, включая транспорт, сетевые системы, социальные сети и научные исследования.*



**MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS**

Ключевые слова: Алгоритм Флойда-Уоршелла, кратчайший путь, теория графов, транспортная система, анализ сети, социальная сеть, научные исследования, сложность, алгоритмические подходы, дорожная сеть, маршрутизация, программирование, оптимизация, многосистемы, применение в реальной жизни.

Floyd-Uorshal algoritmi, grafdagi barcha juft nuqtalar orasidagi minimal yo'llarni topish uchun mo'ljallangan samarali va keng tarqalgan algoritmdir. Bu algoritm, ayniqsa, barcha juft nuqtalar o'rtasidagi qisqa yo'l masofalarini hisoblashda qo'llaniladi. Floyd-Uorshal algoritmi, qisqacha aytganda, grafikda har bir juft nuqtaning minimal masofasini topishga qaratilgan. Asosiy printsip, boshlang'ich holatdagi masofalarni matrisda ifodalash va keyinchalik bu matrisni iteratsiyalar orqali yangilab borishdan iborat. Bu jarayonda, har bir vertex orasidagi masofa boshqa vertexlar orqali yangilanadi, bu esa barcha juft nuqtalar o'rtasidagi minimal masofalarni aniqlashga yordam beradi.

Algoritmning ishlash printsipi quyidagicha:

Dastlab, har bir juft vertex orasidagi masofa massivida ma'lum bir qiymatlar o'rnatiladi: o'zaro bog'lanmagan nuqtalar uchun cheksiz (∞) qiymat, o'zaro bog'langan nuqtalar uchun esa ular orasidagi to'g'ridan-to'g'ri yo'l uzunligi. Keyin, har bir vertex orqali o'tilishi mumkin bo'lgan eng qisqa yo'lni topish uchun iteratsiya boshlanadi. Har bir iteratsiyada, algoritm grafdagi har bir vertexni ko'rib chiqadi va bu vertexni boshqa har bir vertex bilan bog'lagan yo'lni yangilaydi. Bunda, masofa o'lchovlarida qisqarishlar amalga oshiriladi, ya'ni agar yangi yo'l avvalgi yo'ldan qisqaroq bo'lsa, unda eski yo'l qiymati yangilangan qiymatga o'zgartiriladi. Bu jarayon barcha vertexlar o'rtasidagi minimal masofalar hisoblanganiga qadar davom etadi. Natijada, algoritm barcha juft nuqtalar orasidagi minimal yo'llarni aniqlaydi.

Floyd-Uorshal algoritmi o'zining oddiyligi va samaradorligi bilan ajralib turadi, ammo uning eng katta afzalliklaridan biri, grafikdagi har bir juft nuqtalar o'rtasidagi yo'llarni bir martalik hisoblashidir. Bu algoritmni qo'llashda asosiy e'tibor, har bir iteratsiyada grafdagi barcha vertexlar orasidagi eng qisqa yo'lni yangilashga qaratilgan bo'lib, bu esa uni masofalarni tez va samarali hisoblash imkonini beradi. Shu bilan birga, algoritm grafning barcha juft vertexlari orasidagi minimal yo'llarni hisoblashda ishlatilishi mumkin bo'lib, bu uni katta grafiklarda, masalan, tarmoq marshrutlash, transport tizimlarida yoki ijtimoiy tarmoqlarda foydalidir.

Shu bilan birga, Floyd-Uorshal algoritmidagi boshqa grafik algoritmlaridan farqli ravishda, qo'shimcha holatlarda natijalarning yangilanmasligi yoki yangi yo'l topishning ortiqcha vaqt talab qilishi kabi kamchiliklar yuzaga kelishi mumkin. Biroq, uning ko'plab amaliy qo'llanilishlarida bu kamchiliklar unchalik muhim bo'lmasligi mumkin, chunki algoritm ko'proq yo'llarni bir vaqtning o'zida hisoblashga qaratilgan. Shu sababli, Floyd-Uorshal algoritmi katta va kompleks grafiklarda juda foydalidir, lekin u har doim ham eng samarali algoritm emas.



Floyd-Uorshal algoritmining ishlash printsiplari nafaqat matematik jihatdan, balki dasturlashda ham juda muhimdir. Dasturlashda, bu algoritm ko'pincha to'liq grafda barcha vertexlar orasidagi minimal masofalarni topish uchun ishlatiladi, masalan, marshrutlash protokollarida yoki tarmoqning qayerda eng yaxshi va eng qisqa yo'llar mavjudligini aniqlashda. Agar biror tizimda tarmoq uzilishlari yuzaga kelsa yoki resurslar o'zgarishi kerak bo'lsa, Floyd-Uorshal algoritmi yordamida tezda yangi minimal yo'llarni hisoblash mumkin bo'ladi. Buning natijasida, algoritm juda samarali va tezkor ravishda o'zgartirishlarga moslasha olish imkonini beradi.

Floyd-Uorshal algoritmining asosiy printsiplari, uni dasturlashda va amaliyotda qo'llashda qanday ishlashini tushunishda muhim ahamiyatga ega. Algoritmning ishlash prinsipiga asoslanib, dasturchilar uni turli tarmoqlar va tizimlarda, masalan, tarmoq aloqalarida yoki transport tizimlarida muvaffaqiyatli qo'llashlari mumkin. Shuningdek, uning o'rganish va tahlil qilish osonligi algoritmni yangi boshlovchilar uchun ham foydali bo'ladi. Floyd-Uorshal algoritmining vaqt va xotira murakkabligi uning ishlash samaradorligini va ko'lamini baholashda muhim omil hisoblanadi. Algoritmning vaqt murakkabligi $O(n^3)$ bo'lib, bu uning grafdagi har bir vertex va uning o'zaro aloqalarini ko'rib chiqish jarayoniga asoslanadi. Bu yerda n grafdagi vertexlar sonini anglatadi. Algoritmning har bir iteratsiyasida, ya'ni har bir vertexni tekshirishda, boshqa barcha vertexlar bilan aloqalar o'rganiladi va bu jarayon barcha vertexlar bo'yicha takrorlanadi. Har bir vertexni boshqa vertexlar bilan aloqalarini yangilash uchun $O(n^2)$ vaqt kerak bo'ladi, chunki har bir vertex boshqa barcha vertexlarga bog'langan yoki bog'lanmaganligini tekshiradi. Buning ustiga, har bir vertexni tekshirish jarayoni n marta amalga oshiriladi, natijada umumiy vaqt murakkabligi $O(n^3)$ ga teng bo'ladi.

Floyd-Uorshal algoritmidagi o'zaro aloqalarni yangilash jarayoni, masalan, bir vertexdan boshqa vertexga eng qisqa yo'lni hisoblashda, barcha vertexlarni bir vaqtning o'zida tekshiradi. Shunday qilib, algoritmning asosiy yuki vertikal va gorizontal tekshiruvlarning birgalikda amalga oshirilishidan kelib chiqadi. Bu jarayon uchun umumiy vaqt murakkabligini $O(n^3)$ deb baholash mumkin, chunki har bir vertexni tekshirish uchun n marta iteratsiya qilish kerak bo'ladi. Shu sababli, Floyd-Uorshal algoritmi katta grafalar bilan ishlashda samarali bo'lishi mumkin, ammo uning ishlash tezligi katta grafalarda sezilarli darajada pasayadi.

Algoritmning xotira murakkabligi ham $O(n^2)$ ga teng, chunki grafning barcha vertexlari orasidagi aloqalar uchun maxsus matritsa tuziladi. Har bir juft vertex orasidagi masofani saqlash uchun $n \times n$ o'lchamdagi matritsa talab qilinadi. Bu matritsa dastlabki holatda to'ldiriladi, masalan, har bir bog'langan vertex o'rtasidagi masofalar o'rnatiladi, bog'lanmagan vertexlar orasida esa ∞ (cheksiz) qiymati saqlanadi. Transport tizimlarida ham Floyd-Uorshal algoritmiga talab katta. Ayniqsa, yo'l tarmog'idagi har bir shahar yoki nuqta orasidagi eng qisqa yo'lni hisoblashda bu algoritmning ahamiyati katta. Agar bir necha shaharlar o'rtasida transport aloqalari bo'lsa, ular orasidagi eng tezkor yo'lni topish uchun Floyd-Uorshal algoritmi qo'llanadi. Misol uchun, shaharlarga temir yo'l



MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

yoki avtomobil yo'llari orqali qanday qilib eng qisqa va eng arzon yo'l bilan yetib borishni aniqlashda, bu algoritmi barcha imkoniyatlarni tekshirib, eng maqbul variantni tanlaydi.

Misol**Internet tarmog'ida eng qisqa yo'lni topish**

Bizda bir nechta serverlar (tugunlar) bo'lgan va ularning o'rtasida aloqalar (yo'llar) mavjud. Har bir yo'lning o'tish vaqti (yoki masofasi) berilgan. Floyd-Uorshal algoritmi yordamida tarmoqdagi barcha serverlar o'rtasidagi eng qisqa yo'llarni hisoblaymiz. Quyidagi misolda, serverlar orasidagi aloqalar va ularning masofalari (yoki vaqtlarini) berilgan bo'ladi.

Grafni tasvirlash:

Tugunlar: serverlar (A, B, C, D)

Ularining o'rtasidagi masofalar:

A -> B: 5

A -> C: 10

B -> D: 2

C -> D: 1

Algoritmi:

1. Boshlang'ich grafni yaratamiz, ya'ni barcha nuqtalar orasidagi masofalarni saqlovchi matritsani tuzamiz.
2. Floyd-Uorshal algoritmidagi biz har bir nuqta orasidagi eng qisqa yo'lni topish uchun, barcha nuqtalarni bir-biriga bog'lashga harakat qilamiz.
3. Oxirida, har bir juft nuqta o'rtasidagi eng qisqa yo'lni (masofani) hisoblab chiqamiz.

using System;

```
class FloydWarshall
```

```
{
```

```
    static int INF = int.MaxValue;
```

```
    public static void FloydWarshallAlgorithm(int[,] graph, int n)
```

```
    {
```

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

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

```
        {
```

```
            for (int j = 0; j < n; j++)
```

```
            {
```

```
                if (i == j)
```

```
                    dist[i, j] = 0;
```

```
                else if (graph[i, j] != 0)
```

```
                    dist[i, j] = graph[i, j];
```

```
            }
```

```
        }
```

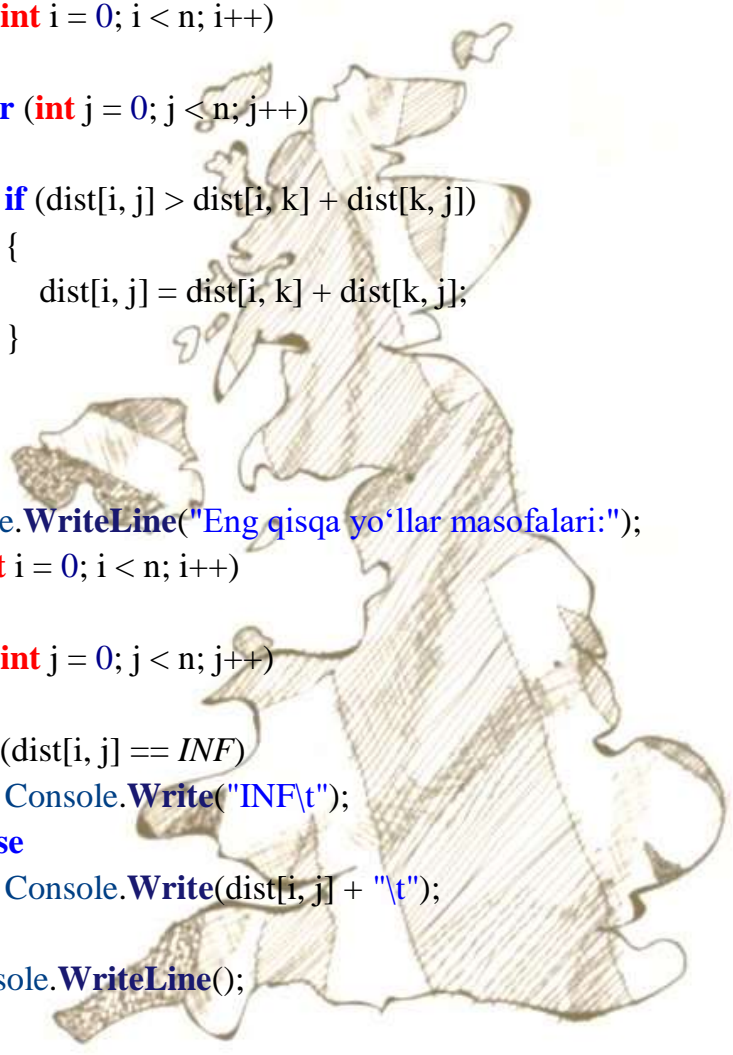
```
    }
```



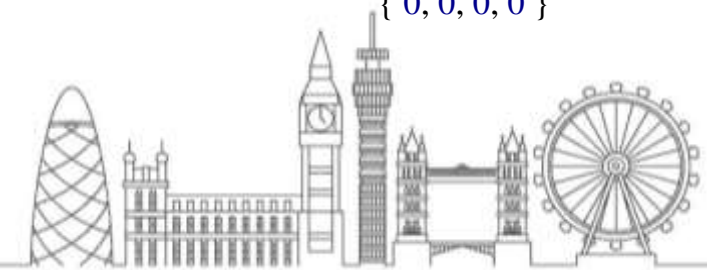


MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

```
else
    dist[i, j] = INF;
}
}
for (int k = 0; k < n; k++)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (dist[i, j] > dist[i, k] + dist[k, j])
            {
                dist[i, j] = dist[i, k] + dist[k, j];
            }
        }
    }
}
Console.WriteLine("Eng qisqa yo'llar masofalari:");
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        if (dist[i, j] == INF)
            Console.WriteLine("INF\t");
        else
            Console.WriteLine(dist[i, j] + "\t");
    }
    Console.WriteLine();
}
}
```



```
static void Main()
{
    int n = 4;
    int[,] graph = new int[,]
    {
        { 0, 5, 10, 0 },
        { 0, 0, 0, 2 },
        { 0, 0, 0, 1 },
        { 0, 0, 0, 0 }
    }
}
```





```
};  
FloydWarshallAlgorithm(graph, n);  
}  
}
```

Natija:

Eng qisqa yo‘llar masofalari:

0 5 10 INF
INF 0 INF 2
INF INF 0 1
INF INF INF 0

Bu natijada:

A dan B gacha masofa 5.

A dan C gacha masofa 10.

B dan D gacha masofa 2.

C dan D gacha masofa 1.

Algoritmda cheksiz masofa (INF) bo‘lsa, bu nuqtalar orasida to‘g‘ridan-to‘g‘ri yo‘l mavjud emasligini anglatadi.

Floyd-Uorshal algoritmi, o‘zining soddaligi va samaradorligi bilan turli xil amaliyotlarda keng qo‘llaniladi. Uning asosiy vazifasi grafdagi barcha juft nuqtalar orasidagi minimal masofalarni aniqlashdir, bu esa transport tizimlaridan tortib, ijtimoiy tarmoqlar va ilmiy tadqiqotlarga bo‘lgan sohalarda muhim ahamiyatga ega. Algoritmnining tarmoq, logistik, ijtimoiy aloqalar, va ko‘p agentli tizimlar kabi turli sohalardagi qo‘llanilishi uning amaliyotdagi ahamiyatini yanada oshiradi. Shuningdek, uning vaqt va xotira murakkabligi va turli yondashuvlari uni keng miqyosda optimallashtirish imkoniyatlarini taqdim etadi. Umuman olganda, Floyd-Uorshal algoritmi – bu samarali, keng qo‘llaniladigan va murakkab tizimlarni boshqarish va optimallashtirishda muhim rol o‘ynaydigan bir vositadir.

FOYDALANILGAN ADABIYOTLAR:

1. Floyd, R. W. (1962). *Algorithm 97: Shortest Path*. Communications of the ACM, 5(6), 345.
2. Warshall, S. (1962). *The Algorithmic Problem of Finding the Shortest Path*. Journal of the ACM, 9(1), 15-28.
3. Dijkstra, E. W. (1959). *A Note on Two Problems in Connexion with Graphs*. Numerische Mathematik, 1, 269-271.
4. Tarjan, R. E. (1983). *Data Structures and Network Algorithms*. SIAM.
5. Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.
6. Kleinberg, J., & Tardos, É. (2006). *Algorithm Design*. Pearson.





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

7. Skiena, S. S. (2008). *The Algorithm Design Manual* (2nd ed.). Springer.
8. Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
9. Hsu, W., & Hwang, C. (1994). *A Survey of Shortest Path Algorithms*. *Computers and Operations Research*, 21(3), 305-314.
10. Miettinen, M., & Møller, P. (2004). *Solving Shortest Path Problems with Multiple Constraints*. *European Journal of Operational Research*, 154(3), 723-735.
11. Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.
12. 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.
13. Farmonov, S., & Ergashaliyeva, B. (2024). QAT'YMAS NEYRON TO'RLAR: MAMDANI QAT'YMAS MANTIQIY XULOSASI, SUGENO QAT'YMAS MANTIQIY XULOSASI. *Development and innovations in science*, 3(5), 62-70.
14. Farmonov, S., & Rustamova, N. (2024, May). SINFLASHNING METRIK ALGORITMLARI, YAQIN QO'SHNI USULI VA UNI UMUMLASHTIRISH HAMDA ULARNI NEYRON TARMOQ TEXNOLOGIYALARIDA QO'LLANILISHI. In *Международная конференция академических наук* (Vol. 3, No. 5, pp. 71-75).
15. Raxmonjonovich, F. S. (2024). DINAMIK DASTURLASH VA TARMOQ OQIMIDA FORD-BELMAN ALGORITMIDAN FOYDALANISH. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 58(7), 13-19.
16. Raxmonjonovich, F. S. (2024). IJTIMOIY TARMOQLAR TAHLILIDA BFS ALGORITMLARI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 58(7), 20-26.
17. Raxmonjonovich, F. S., & Xurshidbek o'g'li, A. O. (2024). FORD-BELMAN ALGORITMI. *Modern education and development*, 15(4), 60-65.
18. Raxmonjonovich, F. S. (2024). KOMPYUTER TARMOQLARI SOHASIDA BITLI ALGORITMLAR. *Modern education and development*, 15(4), 50-59.
19. Raxmonjonovich, F. S. (2024). BIR SHAHARDAN BOSHQASIGA YUK YETKAZIB BERISHDA ENG OPTIMAL VA KAM XARAJAT SARFLANADIGAN YO'LNI TOPISHDA BELLMAN-FORD ALGORITMIDAN FOYDALANISH. *Ta'lim innovatsiyasi va integratsiyasi*, 34(2), 72-78.
20. Raxmonjonovich, F. S. (2024). AXBOROTLARNI SHIFRLASHDA MATEMATIK ALGORITMLARDAN FOYDALANISH. *Modern education and development*, 15(5), 338-344.
21. Farmonov, S. R., & qizi Oktamjonova, M. I. (2024, November). FLOYD-UORSHELL ALGORITMI. In *International Conference on World Science and Resarch* (Vol. 1, No. 3, pp. 32-42).





MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS

22. Farmonov, S. R. (2024). BFS ALGORITMI ORQALI TOPOLOGIK TARTIBNI ANIQLASH. ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ, (56-5).

23. Raxmonjonovich, F. S. (2024). KAN ALGORITMI ASOSIDA GRAFLARDA SIKLSIZ TARTIBNI ANIQLASH. Modern education and development, 15(4), 43-49.

24. Raxmonjonovich, F. S. (2024). JONSON ALGORITMI BILAN BOG 'LIQ MUAMMOLAR VA ULARNI YECHISH USULLARI. Modern education and development, 15(4), 32-42.

