

**FORD-BELLMAN ALGORITIMINI TARMOQ VA TRANSPORT
MUAMMOLARIGA TADBIG'I****Farmonov Sherzodbek Raxmonjonovich***Farg'ona davlat universiteti amaliy matematika va informatika**kafedrası katta o'qituvchisi**farmonovsh@gmail.com***Tojialiyeva Mohigul Muzrobjon qizi***Farg'ona davlat universiteti 2-kurs talabasi**mohigultojialiyeval0@gmail.com*

Annotatsiya. Ushbu maqolada Ford-Bellman algoritmi va uning tarmoq va transport tizimlaridagi qo'llanilishi ko'rib chiqilgan. Ford-Bellman algoritmi grafda manfiy og'irliklar mavjud bo'lsa ham, boshlang'ich tugundan barcha tugunlarga eng qisqa yo'lni topish imkonini beradi. Maqolada algoritmning ishlash prinsipi, uning tarmoqda ma'lumot uzatish, internet yo'naltirish va transport tizimlaridagi roli ko'rsatilgan. Shuningdek, C# dasturlash tilida Ford-Bellman algoritmining implementatsiyasi va transport tizimi misoli keltirilgan. Maqola tarmoq va transport tizimlarida resurslardan samarali foydalanish va optimallashtirishdagi ahamiyatini ta'kidlaydi.

Kalit so'zlar. Ford-Bellman algoritmi, eng qisqa yo'l, Transport muammolari, Algoritmlar, c# dasturlash tili, Manfiy og'irlik, Logistika.

Abstract. This article reviews the Ford-Bellman algorithm and its applications in network and transportation systems. The Ford-Bellman algorithm allows finding the shortest path from the starting node to all nodes, even if there are negative weights in the graph. The article shows the principle of the algorithm, its role in network data transmission, Internet routing and transport systems. Also, the implementation of the Ford-Bellman algorithm in the C# programming language and an example of a transport system are presented. The article emphasizes the importance of efficient use and optimization of resources in network and transport systems.

Key words. Ford-Bellman algorithm, shortest path, Transportation problems, Algorithms, c# programming language, Negative weight, Logistics.

Абстрактный. В этой статье рассматривается алгоритм Форда-Беллмана и его применение в сетевых и транспортных системах. Алгоритм Форда-Беллмана позволяет найти кратчайший путь от начального узла ко всем узлам, даже если в графе имеются отрицательные веса. В статье показан принцип работы алгоритма, его роль в сетевой передаче данных, интернет-маршрутизации и транспортных системах. Также представлена реализация алгоритма Форда-Беллмана на языке программирования C# и пример транспортной системы. В статье подчеркивается важность эффективного использования и оптимизации ресурсов сетевых и транспортных систем.

Ключевые слова. Алгоритм Форда-Беллмана, кратчайший путь, Транспортные задачи, Алгоритмы, язык программирования C#, Отрицательный вес, Логистика.

Ford-Bellman Algoritmi: Asosiy Tushunchalar

Ford-Bellman algoritmi 1958 yilda Richard Bellman va Lester R. Ford tomonidan ishlab chiqilgan bo'lib, uning asosiy maqsadi grafda manfiy og'irlikli qirralar mavjud bo'lsa ham, boshlang'ich tugundan boshqa barcha tugunlarga eng qisqa yo'lni aniqlashdir. Algoritm, shuningdek, grafda manfiy sikl mavjudligini aniqlash imkoniyatiga ega.

Algoritm quyidagi bosqichlarda ishlaydi:

1. **Boshlang'ich qiymatlarni o'rnatish:** Har bir tugun uchun boshlang'ich masofa qiymatini ∞ (cheksiz) deb belgilaymiz, faqat boshlang'ich tugunning qiymatini 0 qilib belgilaymiz.

2. **Iteratsiyalarni amalga oshirish:** Algoritm barcha qirralarni ko'rib chiqadi va tugunlar orasidagi eng qisqa masofani yangilaydi. Bu jarayon $|V|-1$ (tugunlar soni - 1) marta takrorlanadi.

3. **Manfiy sikllarni tekshirish:** Agar oxirgi iteratsiyada har qanday yangilanish amalga oshsa, demak, grafda manfiy sikl mavjud.

Ford-Bellman algoritmi ko'plab muammolarni hal qilishda ishlatiladi, ayniqsa, tarmoq va transport tizimlarida, chunki u manfiy og'irliklarga ega bo'lgan qirralarni ham qo'llab-quvvatlaydi.

Ford-Bellman Algoritmining Tarmoq Muammolarida Tadbiqi

Tarmoq muammolarida Ford-Bellman algoritmi, ayniqsa, ma'lumotlarni uzatish va tarmoq yo'naltirish (routing) masalalarini hal qilishda keng qo'llaniladi. Tarmoqdagi tugunlar va qirralar o'rtasida eng qisqa yo'lni topish zarurati mavjud bo'lsa, Ford-Bellman algoritmi eng maqbul yechimlarni taqdim etadi.

A. Tarmoqda Ma'lumotlarni Uzatish

Tarmoqda ma'lumotlarni uzatishda ma'lum bir tugundan boshqa tugunga ma'lumot yuborish kerak bo'lganda, tarmoqdagi eng qisqa yo'lni topish muhimdir. Ford-Bellman algoritmi yordamida, tarmoqdagi barcha tugunlar orasidagi eng qisqa yo'lni hisoblash mumkin. Tarmoqdagi qirralar orasidagi kechikishlar, kengayishlar yoki boshqa omillar grafning og'irligi sifatida hisoblanishi mumkin.

Tarmoq tarmoqlarida (masalan, kompyuter tarmoqlari yoki mobil tarmoqlar) har bir tugun o'zining xususiyatlariga ega bo'lib, ma'lumot uzatish uchun eng yaxshi yo'lni aniqlashda Ford-Bellman algoritmi samarali ishlaydi. Algoritm har bir tugun o'rtasidagi eng qisqa yo'lni aniqlab, tarmoqdagi uzilishlarni oldini olishga yordam beradi.

B. Internetda Yo'naltirish (Routing)

Internet tarmog'ida yo'naltirish protokollari (masalan, RIP – Routing Information Protocol) tarmoqdagi eng qisqa yo'llarni aniqlash uchun Ford-Bellman algoritmidan foydalanadi. Internetda, bir nechta tugunlar (ruterlari) mavjud bo'lib, ular orasida eng qisqa va eng tezkor yo'naltirish yo'lini tanlash uchun bu algoritm qo'llaniladi.

Ford-Bellman algoritmi orqali, tarmoqda eng tezkor yo'naltirish yo'llarini topish, tarmoq resurslaridan samarali foydalanishga yordam beradi. Bu, ayniqsa, o'zgaruvchan va murakkab tarmoq konfiguratsiyalarida muhimdir, chunki har doim tarmoqning qirralari (masalan, tarmoqning kechikish va o'zgaruvchanligi) o'zgarib turadi.

Ford-Bellman Algoritmining Transport Muammolarida Tadbiqi

Ford-Bellman algoritmi transport tizimlarida, masalan, yo'l tarmog'ini optimallashtirish, yuk tashish va shahar ichidagi transportni boshqarish kabi muammolarni hal qilishda samarali ishlatiladi. Transport tizimlarida, yo'llar va ularning og'irliklari (kechikishlar, yo'l to'siqlari) grafning qirralari sifatida ko'rib chiqilishi mumkin.

A. Transport Tizimlarida Yo'lni Optimallashtirish

Transport tizimlarida, masalan, avtomobillar, avtobuslar yoki poezdlar uchun eng qisqa yo'lni topish muhimdir. Ford-Bellman algoritmi, har bir yo'nalishning uzunligini, kechikishlarini va boshqa omillarni hisobga olgan holda, eng qisqa va samarali yo'lni topishda ishlatiladi. Yo'llar orasidagi kechikishlar, to'siqlar va boshqa omillar manfiy og'irliklar sifatida ko'rib chiqiladi.

Shuningdek, shahar ichidagi transport tarmog'ini boshqarishda Ford-Bellman algoritmi avtobuslar, tramvaylar yoki metro tizimlari uchun eng qisqa marshrutlarni aniqlashda ishlatiladi. Bu transport vositalari orasidagi eng qisqa va eng samarali yo'nalishlarni topishda yordam beradi.

B. Logistika va Ta'minot Zanjiri

Ford-Bellman algoritmi logistik tizimlarda ham qo'llaniladi, ayniqsa, yuklarni bir nuqtadan boshqa nuqtaga tashishda eng qisqa va samarali yo'llarni aniqlashda. Yuk tashishda vaqt, masofa va boshqa omillar (masalan, yuk sig'imi) grafning og'irliklari sifatida ishlatiladi. Algoritm bu omillarni hisobga olib, eng maqbul yo'lni topishga yordam beradi.

Transport tarmoqlari va ta'minot zanjirlarida, yuklarni tezda va samarali yetkazib berish uchun eng qisqa va eng arzon yo'llarni aniqlashda Ford-Bellman algoritmi katta yordam beradi.

Cheklovlar va Xulosa

Ford-Bellman algoritmi juda samarali va ko'p sohalarda qo'llaniladigan algoritm bo'lsa-da, ba'zi cheklovlarga ham ega. Agar grafda manfiy sikl mavjud bo'lsa, algoritmning natijasi noaniq bo'lishi mumkin. Shuningdek, uning vaqt murakkabligi $O(V \cdot E)$ bo'lib, katta tarmoqlar va transport tizimlari uchun samarali bo'lmasligi mumkin.

Shunga qaramay, Ford-Bellman algoritmi tarmoq va transport tizimlarida eng qisqa yo'lni topish va muammolarni samarali hal qilishda keng qo'llaniladi. Tarmoqda ma'lumotlarni uzatish, yo'naltirish va transport tizimlarida eng yaxshi yo'llarni aniqlash uchun bu algoritmning foydalari juda katta. Kelajakda, ushbu algoritmni yanada samarali qilish uchun yangi optimallashtirish usullari va parallel hisoblash texnologiyalari qo'llanilishi mumkin.

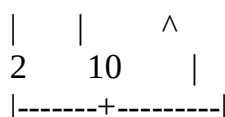
Misol:

Aytaylik, bir transport tarmog'ida ba'zi yo'llar orasida kechikishlar mavjud. Sizning vazifangiz — boshlang'ich nuqtadan (tugundan) boshqa barcha nuqtalarga eng qisqa yo'lni aniqlash. Yo'llar orasidagi kechikishlar (og'irliklar) quyidagicha:

- $A \rightarrow B$: 4
- $A \rightarrow C$: 2
- $B \rightarrow C$: 5
- $B \rightarrow D$: 10
- $C \rightarrow D$: 3

Grafni quyidagicha tasavvur qilaylik:

A --4--> B --5--> C --3--> D



Bu yerda, A boshlang'ich tugun bo'lib, biz Adan barcha boshqa tugunlarga eng qisqa yo'lni topishimiz kerak.

○ **Graph sinfi:** Bu sinfda grafni va uning tugunlarini tashkil etamiz. AddEdge metodi orqali grafga qirralar qo'shiladi. Har bir qirra manzil, manba tugun va og'irlikdan iborat.

○ **BellmanFord metodi:** Bu metod Ford-Bellman algoritmini amalga oshiradi:

1. **Boshlang'ich qiymatlarni o'rnatish:** Har bir tugun uchun masofa int.MaxValue deb belgilanadi, faqat boshlang'ich tugunning qiymati 0 qilib belgilanadi.

2. **Iteratsiyalar:** V-1 marta qirralar ustidan aylanib, eng qisqa yo'llarni yangilab boramiz.

3. **Manfiy siklni tekshirish:** Agar oxirgi iteratsiyalarda biron bir yo'l o'zgartirilgan bo'lsa, bu grafda manfiy sikl borligini bildiradi.

4. **Natijalar:** Oxirida har bir tugunga eng qisqa yo'lni chiqaramiz.

C# tilida Ford-Bellman algoritmini quyidagi tarzda implementatsiya qilamiz.

```

class                                     Program
{
    public                               class                               Edge
    {
        public           int           Source           {           get;           set;           }
        public           int           Destination       {           get;           set;           }
        public           int           Weight            {           get;           set;           }

        public           Edge(int       source,           int       destination,           int       weight)
        {
            Source       = source;
            Destination   = destination;
            Weight        = weight;
        }
    }
}

public                               class                               Graph
{

```

```

public class Graph {
    private List<Edge> Edges;
    private int vertices;

    public Graph(int vertices) {
        this.vertices = vertices;
        Edges = new List<Edge>();
    }

    public void AddEdge(int source, int destination, int weight) {
        Edges.Add(new Edge(source, destination, weight));
    }

    public void BellmanFord(int start) {
        // 1. Boshlang'ich masofalar
        int[] dist = new int[V];
        for (int i = 0; i < V; i++) {
            dist[i] = int.MaxValue;
        }
        dist[start] = 0;

        // 2. Eng qisqa yo'llarni hisoblash
        for (int i = 1; i < V; i++) { // V - 1 marta iteratsiya
            foreach (var edge in Edges) {
                if (dist[edge.Source] != int.MaxValue && dist[edge.Source] + edge.Weight < dist[edge.Destination]) {
                    dist[edge.Destination] = dist[edge.Source] + edge.Weight;
                }
            }
        }

        // 3. Manfiy siklni tekshirish
        foreach (var edge in Edges) {
            if (dist[edge.Source] != int.MaxValue && dist[edge.Source] + edge.Weight <

```

```

dist[edge.Destination])
    {
        Console.WriteLine("Grafda          manfiy          sikl          mavjud!");
        return;
    }
}

//          4.          Natijalarni          chiqarish
Console.WriteLine("Boshlang'ich tugundan (A) barcha tugunlarga eng qisqa
yo'llar:");
for (int i = 0; i < V; i++)
{
    if (dist[i] == int.MaxValue)
    {
        Console.WriteLine($"Tugun {i}:          Ulgurmadik");
    }
    else
    {
        Console.WriteLine($"Tugun {i}:          {dist[i]}");
    }
}
}

static void Main()
{
    //          5          ta          tugun          (A,          B,          C,          D)
    Graph graph = new Graph(4);

    //          Yo'llarni          qo'shish          (qirralar)
    graph.AddEdge(0, 1, 4); //          A          ->          B:          4
    graph.AddEdge(0, 2, 2); //          A          ->          C:          2
    graph.AddEdge(1, 2, 5); //          B          ->          C:          5
    graph.AddEdge(1, 3, 10); //          B          ->          D:          10
    graph.AddEdge(2, 3, 3); //          C          ->          D:          3

    //          Bellman-Ford algoritmini chaqirish (Boshlang'ich tugun A (0))
    graph.BellmanFord(0);
}
}
Natija:

```

Boshlang'ich tugundan (A) barcha tugunlarga eng qisqa yo'llar:

Tugun 0: 0

Tugun 1: 4

Tugun 2: 2

Tugun 3: 5

Xulosa: Ford-Bellman algoritmi juda samarali va ko'p sohalarda qo'llaniladigan algoritm bo'lsa-da, ba'zi cheklovlariga ham ega. Agar grafda manfiy sikl mavjud bo'lsa, algoritmnining natijasi noaniq bo'lishi mumkin. Shuningdek, uning vaqt murakkabligi $O(V \cdot E)$ bo'lib, katta tarmoqlar va transport tizimlari uchun samarali bo'lmasligi mumkin.

FOYDANILGAN ADABIYOTLAR:

1. Bellman, R. (1958). "On a routing problem." *Quarterly of Applied Mathematics*, 16(1), 87–90.
2. Ford, L. R., & Fulkerson, D. R. (1956). "Maximal flow through a network." *Canadian Journal of Mathematics*, 8(3), 399–404.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
4. Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs." *Numerische Mathematik*, 1, 269–271.
5. Knuth, D. E. (1973). *The Art of Computer Programming, Volume 1: Fundamental Algorithms* (3rd ed.). Addison-Wesley.
6. Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall.
7. Bertsekas, D. P. (1998). *Nonlinear Programming* (2nd ed.). Athena Scientific.
8. Liu, Z., & Hu, Z. (2005). "An Improved Bellman-Ford Algorithm for Shortest Path Problems." *Journal of Computational Information Systems*, 1(6), 32–37.
9. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. B CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (T. 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). OBEKTGA YO'NALTIRILGAN DASTURLASH ZAMONAVIY DASTURLASHNING ASOSI SIFATIDA. *Центральноазиатский 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.

24. Farmonov, S. R., & qizi Xomidova, M. A. (2024). C# VA JAVA DASTURLASH TILLARIDA FAYLLAR BILAN ISHLASHNING TURLI USULLARINING SAMARADORLIGI HAQIDA. *Zamonaviy fan va ta'lim yangiliklari xalqaro ilmiy jurnal*, 1(9), 45-51.

25. Raxmonjonovich, F. S. (2024). C# VA MASHINA TILI. *Ta'lim innovatsiyasi va integratsiyasi*, 12(1), 59-62.