

FORD-BELMAN ALGORITMINI ENG QISQA YO'LNI TOPISH MASALASIDA QO'LLANILISHI

Farmonov Sherzodbek Raxmonjonovich

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

farmonovsh@gmail.com

Rahimova Zarifaxon Shuxratjon qizi

Farg'ona davlat universiteti talabasi

zarifaxonrahimova04@gmail.com

Annotatsiya. Ushbu maqolada Ford-Bellman algoritmi, grafik nazari va optimizatsiya sohalarida eng qisqa yo'lni topish uchun ishlatiladigan muhim algoritm sifatida ko'rib chiqiladi. Algoritmning asosiy tamoyillari, ishlash prinsipi, afzalliklari va kamchiliklari, shuningdek, uning amaliy qo'llanilishi tahlil qilinadi. Ford-Bellman algoritmi manfiy og'irliklarga ega grafiklar bilan ishlash qobiliyati bilan ajralib turadi, bu esa uni Dijkstra algoritmidan farqlaydi. Maqola, algoritmning transport tizimlari, telekommunikatsiya, iqtisodiyot va o'yinlar kabi sohalaridagi amaliy qo'llanilishini ko'rsatadi. Ford-Bellman algoritmining samaradorligini oshirish va yangi yondashuvlar ishlab chiqish istiqbollari ham muhokama qilinadi.

Kalit so'zlar. Ford-Bellman algoritmi Eng qisqa yo'l, grafik nazari, manfiy og'irliklar, transport tizimlari, telekommunikatsiya, iqtisodiy optimizatsiya, o'yinlar, algoritmning afzalliklari, algoritmning kamchiliklari

Annotation. This article discusses the Ford-Bellman algorithm as an important algorithm used to find the shortest path in graph theory, optimization, and related fields. The fundamental principles of the algorithm, its working mechanism, advantages and disadvantages, as well as its practical applications, are analyzed. The Ford-Bellman algorithm is distinguished by its ability to work with graphs that have negative weights, setting it apart from the Dijkstra algorithm. The article demonstrates its practical applications in fields such as transportation systems, telecommunications, economics, and games. The prospects for improving the efficiency of the Ford-Bellman algorithm and developing new approaches are also discussed.

Key words. Ford-Bellman algorithm, Shortest path, Graph theory, Negative weights, Transportation systems, Telecommunications, Economic optimization, Games, Advantages of the algorithm, Disadvantages of the algorithm

Аннотация. В этой статье рассматривается алгоритм Форда-Беллмана как важный алгоритм для нахождения кратчайшего пути в графах, теории графов и оптимизации. Анализируются основные принципы алгоритма, его принцип работы, преимущества и недостатки, а также его практическое применение. Алгоритм Форда-Беллмана выделяется своей способностью работать с графами, имеющими отрицательные веса, что отличает его от алгоритма Дейкстры. В статье показано его практическое применение в таких областях, как транспортные системы, телекоммуникации, экономика и игры. Также обсуждаются перспективы повышения эффективности алгоритма Форда-Беллмана и разработки новых подходов.

Ключевые слова. Кратчайший путь, Теория графов, Отрицательные веса, Транспортные системы, Телекоммуникации, Экономическая оптимизация, Игры, Преимущества алгоритма, Недостатки алгоритма

Ford-Bellman algoritmi, yoki Bellman-Ford algoritmi, grafik nazari va optimizatsiya sohalarida keng qo'llaniladigan muhim algoritmlardan biridir. U, asosan, og'irliklari ijobiy yoki manfiy

bo'lishi mumkin bo'lgan yo'llar orqali grafiklarda eng qisqa yo'lni topish uchun ishlatiladi. Ushbu maqolada Ford-Bellman algoritmining asosiy tamoyillari, ishlash prinsipi, afzalliklari va kamchiliklari, shuningdek, uning amaliy qo'llanilishi ko'rib chiqiladi.

Ford-Bellman algoritmi, 1958 yilda Richard Bellman va Lester Ford tomonidan ishlab chiqilgan. Algoritmning asosiy maqsadi, berilgan grafikda manba tugunidan (vertex) boshqa barcha tugunlarga eng qisqa yo'lni topishdir. Algoritmning asosiy tamoyillari quyidagilardan iborat:

1. Grafikning Strukturasi: Algoritm yo'llar va tugunlardan iborat grafikda ishlaydi. Har bir yo'lning og'irligi (masofasi) bo'lishi mumkin, bu esa ijobiy yoki manfiy qiymatga ega bo'lishi mumkin.

2. Iteratsion Yondashuv: Algoritm, har bir tugun uchun eng qisqa yo'lni topish uchun iteratsion jarayonni amalga oshiradi. Har bir iteratsiyada, tugunlar orasidagi yo'llar ko'rib chiqiladi va eng qisqa yo'l yangilanadi.

3. Manfiy Tsikllar: Ford-Bellman algoritmi manfiy og'irliklarga ega yo'llar bilan ishlay oladi, bu esa uni Dijkstra algoritmidan ajratib turadi. Agar grafikda manfiy tsikl mavjud bo'lsa, algoritm bu holatni aniqlay oladi.

Ford-Bellman algoritmi quyidagi qadamlar orqali amalga oshiriladi:

1. Boshlang'ich Shartlar: Manba tugunidan boshqa barcha tugunlarga masofani cheksiz (∞) deb belgilash va manba tuguniga masofani 0 deb belgilash.

2. Iteratsiyalar: Grafikdagi barcha yo'llarni ko'rib chiqish va har bir tugun uchun eng qisqa yo'lni yangilash. Bu jarayon tugunlar soni minus bir marta takrorlanadi.

3. Manfiy Tsikllarni Tekshirish: Agar biron bir tugun uchun masofa yangilanadigan bo'lsa, bu grafikda manfiy tsikl mavjudligini anglatadi.

Algoritmning afzallik va kamchiliklari

Afzalliklari:

Manfiy og'irliklarga ega grafiklar bilan ishlay oladi.

Oddiy va tushunarli tuzilishga ega.

Oson implementatsiya qilinadi.

Kamchiliklari:

Algoritmning vaqt murakkabligi $O(VE)$ (V — tugunlar soni, E — yo'llar soni), bu esa katta grafiklar uchun sekin ishlashiga olib kelishi mumkin.

Dijkstra algoritmiga nisbatan samaradorligi pastroq.

Ford-Bellman algoritmi ko'plab sohalarda qo'llaniladi, jumladan:

1. Transport Tizimlari: Yo'l tarmoqlarida eng qisqa yo'lni aniqlashda.

2. Telekommunikatsiya: Ma'lumotlar paketlarini eng qisqa yo'l orqali uzatishda.

3. Iqtisodiyot: Resurslarni taqsimlash va optimallashtirishda.

4. O'yinlar: O'yin ichidagi xaritalarda eng qisqa yo'lni topish uchun.

Masala:Grafda berilgan har bir yo'lning og'irligi bilan eng qisqa yo'lni hisoblash. Grafning uchta tuguni va ular orasidagi yo'llar va og'irliklari quyidagicha:

1. Tugunlar: A, B, C

2. Yo'llar va og'irliklar:

A -> B: 4

A -> C: 5

B -> C: -2

3. Boshlang'ich tugun: A

Grafga ko'ra, A tugundan B va C tugunlariga eng qisqa yo'lni topish kerak.

C# kod:

```
public class Programm
```

```
{
    // Ford-Bellman algoritmi
    public static void BellmanFord(int vertices, List<Edge> edges, int start)
    {
        int[] distances = new int[vertices];
        // Boshlang'ich masofalarni cheksiz qilib belgilash
        for (int i = 0; i < vertices; i++)
        {
            distances[i] = int.MaxValue;
        }
        distances[start] = 0;

        // Relaxation jarayonini vertices-1 marta amalga oshiramiz
        for (int i = 1; i < vertices; i++)
        {
            foreach (var edge in edges)
            {
                if (distances[edge.Source] != int.MaxValue && distances[edge.Source] + edge.Weight <
                    distances[edge.Destination])
                {
                    distances[edge.Destination] = distances[edge.Source] + edge.Weight;
                }
            }
        }

        // Salbiy og'irlikli tsiklni tekshirish
        foreach (var edge in edges)
        {
            if (distances[edge.Source] != int.MaxValue && distances[edge.Source] + edge.Weight <
                distances[edge.Destination])
            {
                Console.WriteLine("Grafda salbiy og'irlikli tsikl bor.");
                return;
            }
        }

        // Natijalarni chiqarish
        Console.WriteLine("Eng qisqa masofalar:");
        for (int i = 0; i < vertices; i++)
        {
            if (distances[i] == int.MaxValue)
                Console.WriteLine($"Tugun {i}: Cheksiz");
            else
                Console.WriteLine($"Tugun {i}: {distances[i]}");
        }
    }
}
```

}

Ford-Bellman algoritmi, grafik nazari va optimizatsiya sohalarida muhim ahamiyatga ega. Uning manfiy og'irliklarga ega grafiklar bilan ishlash qobiliyati, uni ko'plab amaliy muammolarni hal qilishda foydali qiladi. Biroq, algoritmning vaqt murakkabligi va samaradorligi cheklovlari mavjud. Kelajakda, Ford-Bellman algoritmini yanada takomillashtirish va yangi yondashuvlar ishlab chiqish, uning amaliy qo'llanilishini kengaytirishi mumkin.

Foydalanilgan adabiyotlar:

1. "Network Flows: Theory, Algorithms, and Applications" - Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin
2. "Optimization Methods in Operations Research and Systems Analysis" - A. Ravindran, D. T. Phillips, J. J. Solberg
3. "Introduction to Operations Research" - Frederick S. Hillier, Gerald J. Lieberman
4. "Operations Research: Applications and Algorithms" - Wayne L. Winston
5. 5. Mykel J. Kochenderfer. Tim A. Wheeler. Algorithms for Optimization. Published by The MIT Press., in London, England. 2019. – 500 p.
6. 6. Рафгарден Тим. Совершенный алгоритм. Графовые алгоритмы и структуры данных. – СПб.: Питер, 2019. - 256 с.
7. 7. Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э. Структуры данных и алгоритмы. – М.: Вильямс, 2018. – 400 с.
8. 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. 9. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. В CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (T. 2, Выпуск 12, сс. 71–74). Zenodo.
10. 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. 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. 12. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 425-429.
13. 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. 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. 15. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 475-481.
16. 16. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 482-486.
17. 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. 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. 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. 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>.