

DINAMIK DASTURLASH VA TARMOQ OQIMIDA FORD-BELMAN ALGORITMIDAN FOYDALANISH

Farmonov Sherzodbek Raxmonjonovich

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

famonovsh@gmail.com

Metinboyeva Fotimaxon Mirqo'zi qizi

Farg'ona Davlat Universiteti 2-kurs talabasi

f34621225@gmail.com

Annotatsiya: Ushbu maqolada Ford-Bellman algoritmining asosiy printsiplari va uning eng qisqa yo'l topish muammolarini hal qilishdagi ahamiyati ko'rib chiqiladi. Algoritmning dinamik dasturlash yondashuvi va manfiy og'irliklarga ega grafalarda ishlash imkoniyati tushuntiriladi. Misollar yordamida algoritmnining ishlashi ko'rsatiladi va uning amaliy qo'llanilishiga e'tibor qaratiladi.

Kalit so'zlar: Ford-Bellman algoritmi, eng qisqa yo'l, dinamik dasturlash, grafning strukturasi, iteratsiya, manfiy aylanishlar, manfiy og'irliklar, tarmoq oqimi, optimal yo'l, yuguruvchi robotlar, o'zgaruvchan muhit, tarmoqni optimallashtirish, foydani maksimal darajaga keltirish, graf nazariyasi, algoritmlar.

Аннотация: В данной статье рассматриваются основные принципы алгоритма Форда-Беллмана и его значимость в решении задач поиска кратчайшего пути. Объясняется подход динамического программирования алгоритма и возможность его применения в графах с отрицательными весами. С помощью примеров демонстрируется работа алгоритма и акцентируется внимание на его практическом применении.

Ключевые слова: Алгоритм Форда-Беллмана, Кратчайший путь, структура графа, итерация, отрицательное вращение, Динамическое программирование, Отрицательные веса, сетевой поток, максимальное использование пользователя, оптимальный способ, беговая дорожка, переменная среда, оптимизация сети, Теория графов, Алгоритмы.

Annotation: This article discusses the fundamental principles of the Ford-Bellman algorithm and its significance in solving shortest path problems. The algorithm's dynamic programming approach and its ability to operate in graphs with negative weights are explained. Examples illustrate the functioning of the algorithm, and its practical applications are highlighted.

Key words: Ford-Bellman algorithm, shortest path, dynamic programming, iteration, the structure of the graph, Negative Weight Cycles, negative weights, Network

Flow, Shortest Path, Robots, Dynamic Environment, Network Optimization, Profit Maximization, graph theory, algorithms.

Ford-Bellman algoritmi, asosan, graf nazariyasi va optimizatsiya sohalarida qo'llaniladigan muhim bir usuldir. U yo'l topish (shortest path) muammolarini hal qilishda keng qo'llaniladi, xususan, og'irliklari musbat yoki manfiy bo'lgan graf strukturalarida. Ushbu maqolada Ford-Bellman algoritmining asosiy tamoyillari, ishlash printsiplari, afzalliklari va cheklovlari haqida batafsil ma'lumot beramiz.

Algoritmining asosiy tamoyillari

Ford-Bellman algoritmi, dastlab 1958-yilda Richard Bellman tomonidan ishlab chiqilgan va keyinchalik L.R. Ford tomonidan takomillashtirilgan. Algoritm, grafda har bir tugun uchun eng qisqa yo'l masofasini hisoblashga qaratilgan. Algoritmining asosiy qoidalari quyidagilardir:

1. **Grafning Strukturalari:** Algoritm yo'llarni hisoblash uchun yo'naltirilgan yoki yo'naltirilmagan grafdan foydalanadi. Har bir qirra (edge) og'irlikka ega bo'lishi mumkin.
2. **Dinamika:** Algoritm dinamik dasturlash usullaridan foydalanadi, ya'ni masalani kichikroq va sodda bo'lgan qismalarga ajratish orqali yechim topadi.
3. **Iteratsiya:** Algoritm har bir tugun uchun eng qisqa yo'lni takroriy ravishda yangilab boradi, bu jarayon tugunlar soni minus bir marta ($V-1$) takrorlanadi.

Algoritmining ishlash printsiplari

Ford-Bellman algoritmining ishlash jarayoni quyidagi bosqichlardan iborat:

1. **Boshlang'ich shartlar:** Barcha tugunlar uchun boshlang'ich masofa (distance) qiymatini cheksiz (∞) deb belgilash va manba tuguni uchun 0 qiymatini o'rnatish.
2. **Yangilash:** Har bir qirra uchun, agar tugun A dan tugun B ga o'tishda yangi masofa (A dan B ga o'tish og'irligi + A dan boshlang'ich tugunga bo'lgan masofa) eski masofadan kichik bo'lsa, masofani yangilang.
3. **Takrorlash:** 2-bosqichni barcha qirralar uchun $V-1$ marta takrorlang (V — grafda tugunlar soni).
4. **Manfiy Tsiklni Tekshirish:** Agar yana bir marta yangilanish mumkin bo'lsa, demak, grafda manfiy tsikl mavjud.

Ford-Bellman algoritmi quyidagi boshqa masalalarda ham qo'llanilishi mumkin:

1. **Manfiy aylanishlar (Negative Weight Cycles) masalalari:** Ford-Bellman algoritmi, grafda manfiy aylanishlar mavjud bo'lsa, ularni aniqlashda ham ishlatiladi. Agar algoritm grafni tahlil qilishda n -chi takrorlashdan so'ng hali ham yo'lni yangilashni davom ettirsa, demak, grafda manfiy og'irlikli aylanish (negative weight cycle) mavjud.

2. Динамик dasturlash va tarmoq oqimi: Ford-Bellman algoritmi ba'zi tarmoq oqimi masalalarida, ayniqsa, "shortest path" hisoblashda, dinamik dasturlash metodlariga asoslangan muammolarni hal qilishda ishlatiladi. Masalan, resurslarni taqsimlash yoki taqsimot tarmoqlari uchun optimal yo'llarni aniqlashda.

3. Yuguruvchi robotlar va o'zgaruvchan muhitlarda navigatsiya: Robotlar yoki boshqa avtomatik tizimlar uchun, ularning harakatini optimallashtirishda, masalan, maqsadli joyga eng qisqa va xavfsiz yo'lni topish uchun Ford-Bellman algoritmi ishlatiladi. Bunda, muhitdagi o'zgaruvchan to'siqlar va og'irliklar (masalan, to'siqlar yoki xavfli zonalar) hisobga olinadi.

4. Yirik ma'lumotlar va tarmoqni optimallashtirish: Internetdagi katta tarmoq tizimlarida, masalan, serverlar orasidagi eng tezkor ma'lumot yuborish yo'lini tanlashda Ford-Bellman algoritmi ishlatiladi. Tarmoqni optimallashtirishda manfiy o'tish vaqtlari yoki tarmoq qiyinchiliklarini hisobga olish muhim.

5. Yuqori darajadagi optimallashtirish va iqtisodiy modellar: Keng miqyosdagi iqtisodiy tizimlarda, masalan, resurslarni ishlab chiqarishdan iste'molga bo'lgan optimal yo'llar yoki xarajatlarni minimallashtirish masalalarida ham Ford-Bellman algoritmi qo'llanilishi mumkin. Bu kabi masalalarda tarmoqdagi barcha nuqtalar o'rtasida eng qisqa va samarali yo'llarni hisoblash kerak bo'lishi mumkin.

6. Qaror qabul qilish va tahlil qilish: Qarorlar qabul qilish jarayonida, ayniqsa, strategik rejalashtirish va optimallashtirishda, resurslar yoki imkoniyatlar bo'yicha eng optimal yo'nalishlarni aniqlashda Ford-Bellman algoritmi yordam berishi mumkin.

7. Shaxsiy moliyaviy va investitsiya tahlili: Moliyaviy sohada, masalan, investitsiya portfelini diversifikatsiya qilishda yoki kreditlar bo'yicha qarorlar qabul qilishda, eng qisqa va eng samarali yo'llar yordamida tahlil qilishda foydalanish mumkin.

Динамик dasturlash va tarmoq oqimi – bu ikkita alohida, lekin bir – biriga yaqin algoritmik tushunchalar bo'lib, ko'plab masalalarni samarali yechishda qo'llaniladi.

1. Динамик dasturlash (Dynamic Programming). Динамик dasturlash (DP) – bu optimal yechimni topishda foydalaniladigan yondashuvdir. U masalalarni kichikroq, bir – biriga o'xshash bo'laklarga bo'lib, har bir bo'lakni bir marta yechib, natijalarni saqlashga asoslanadi. Bu metod ko'plab takrorlanuvchi hisoblashlarni optimallashtirishga yordam beradi.

DP ning asosiy xususiyatlari:

- Optimal substrukturasi: Masalaning yechimi kichik masalalarning optimal yechimlariga asoslanadi.

- Takrorlanadigan subproblemlar: Kichik masalalarning yechimlari bir necha marta ishlatilishi mumkin.

Misol. Fibonacci sonlari. Fibonacci sonlarini hisoblash uchun rekursiv yondashuv ishlatish mumkin, lekin bu juda samarali emas, chunki ko'plab hisoblashlar takrorlanadi.

Rekursiv formula:

$$F(n) = F(n-1) + F(n-2)$$

C # tilidagi kodi:

```
using System;
class Program
{
    static int Fibonacci(int n)
    {
        int[] dp = new int[n + 1];
        dp[0] = 0;
        dp[1] = 1;
        for (int i = 2; i <= n; i++)
        {
            dp[i] = dp[i - 1] + dp[i - 2];
        }
        return dp[n];
    }
    static void Main()
    {
        int n = 10;
        Console.WriteLine("Fibonacci soni: " + Fibonacci(n));
    }
}
```

Natija:

Fibonacci soni: 55

Dasturda Program nomli sinf mavjud. Sinf ichida ikkita metod bor:

Fibonacci - Fibonacci sonini hisoblaydigan metod.

Main - Dastur boshlanishi, unda metod chaqirilib, natija ekranga chiqariladi.

Fibonacci metodini tushuntirish.

Bu metod Fibonacci sonini hisoblaydi. Fibonacci sonlari - bu seriya, unda har bir son oldingi ikkita sonning yig'indisi bo'ladi. Masalan, 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 va h.k.

static int Fibonacci(int n):

static: Bu metodning sinfga bog'lanishini bildiradi. Bu metodni sinfni obyeksiz chaqirishingiz mumkin.

int n: Bu metodga argument sifatida kiritilgan son. Bu son, biz qaysi Fibonacci sonini hisoblamochi ekanligimizni bildiradi (masalan, $n = 10$).

```
int[] dp = new int[n + 1];
```

Bu yerda dp degan massiv yaratiladi, u $n+1$ uzunlikda bo'ladi. Massivda Fibonacci sonlarini saqlaymiz.

dp[0] ni 0, dp[1] ni 1 deb belgilaymiz, chunki Fibonacci ketma-ketligida birinchi ikkita son shunday.

```
dp[0] = 0; dp[1] = 1;
```

Fibonacci sonlarining dastlabki ikkita qiymatini belgilash: dp[0] = 0 va dp[1] = 1.

```
for (int i = 2; i <= n; i++):
```

Bu tsikl 2-chi sonlardan boshlanib, n-chi songacha davom etadi. Har bir bosqichda Fibonacci soni hisoblanadi.

```
dp[i] = dp[i - 1] + dp[i - 2];
```

Bu qatorda, i-chi Fibonacci soni oldingi ikki sonning yig'indisi sifatida hisoblanadi: dp[i - 1] va dp[i - 2].

```
return dp[n];
```

Bu kod satri, kerakli n-chi Fibonacci sonini qaytaradi (ya'ni dp[n]).

Main metodini tushuntirish

```
static void Main():
```

Dastur boshlanishi bo'lib, bunda asosan metodlar chaqiriladi va dastur ishlaydi.

```
int n = 10;
```

Bu qatorda n o'zgaruvchisi e'lon qilinadi va unga qiymat sifatida 10 beriladi. Bu degani, biz 10-chi Fibonacci sonini hisoblashni xohlaymiz.

```
Console.WriteLine("Fibonacci soni: " + Fibonacci(n));
```

Bu yerda Fibonacci(n) metodini chaqiramiz va n ni (ya'ni 10) parametr sifatida uzatamiz. Natijada 10-chi Fibonacci soni hisoblanadi va ekranga chiqariladi.

Fibonacci(n) metodining qaytargan qiymatini Console.WriteLine yordamida chiqaramiz.

Algoritmning afzalliklari:

- **Manfiy Og'irliklarga E'tibor:** Ford-Bellman algoritmi manfiy og'irliklarga ega qirralarni hisoblash imkonini beradi.

- **Oson implementatsiya:** Algoritmni dasturlash oson va tushunarli.

- **Keng qo'llanilish:** Transport tizimlari, tarmoq optimizatsiyasi va iqtisodiy modellashtirish kabi ko'plab sohalarda qo'llaniladi.

Cheklovlar:

- **Tezlik:** Algoritmning vaqt murakkabligi $O(VE)$ ga teng, bu esa katta grafiklar uchun sekin ishlashiga olib kelishi mumkin.

• **Manfiy sikllar:** Agar grafda manfiy tsikllar mavjud bo'lsa, algoritm to'g'ri natija bermaydi va bunday holatlarni aniqlash zarurati paydo bo'ladi.

Ford-Bellman algoritmi eng qisqa yo'l topish muammolarini hal qilishda muhim rol o'ynaydi. Uning dinamik dasturlash asosidagi yondashuvi va manfiy og'irliklarga ega grafalarda ishlash imkoniyati uni juda foydali qiladi. Biroq, uning cheklovlari va sekinligi katta grafiklar bilan ishlashda e'tiborga olinishi kerak. Umuman olganda, Ford-Bellman algoritmi nazariy va amaliy jihatdan muhim ahamiyatga ega bo'lib, ko'plab sohalarda keng qo'llaniladi.

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'KTGA 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.
26. Farmonov, S. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. *Центральноазиатский журнал образования и инноваций*, 2(12 Part 2), 71-74.
27. Farmonov, S., & Jo'rayeva, M. (2023, December). DASTURLASHDA POLIMORFIZMNING AHAMIYATI. In *Международная конференция академических наук* (Vol. 2, No. 13, pp. 5-8).
28. Farmonov, S., & Usmonaliyev, U. (2024). O'ZBEKISTON RESPUBLIKASI IT SOHASINING RIVOJLANISH ISTIQBOLLARI. *Бюллетень педагогов нового Узбекистана*, 2(1), 59-62.
29. Raxmonjonovich, F. S., & Xasan o'g'li, X. O. (2023). DASTURLASHDA SANA VA VAQTLAR BILAN ISHLASH. *Ta'lim innovatsiyasi va integratsiyasi*, 11(11), 3-6.