

BIR SHAHARDAN BOSHQASIGA YUK YETKAZIB BERISHDA ENG OPTIMAL VA KAM XARAJAT SARFLANADIGAN YO'LNI TOPISHDA BELLMAN-FORD ALGORITMIDAN FOYDALANISH

Farmonov Sherzodbek Raxmonjonovich

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

Farmonovsh@gmail.com

Muhammaddiyorova Madina Bahodirjon qizi

Farg'ona davlat universiteti 2-kurs talabasi

Mmuhammaddiyorova0110@gmail.com

Annotatsiya. Ushbu maqolada Bellman-Ford algoritmining qo'llanilish sohalari, ushbu sohalarda aynan qay tarzda foydalanilishi haqida ma'lumot keltirilgan. Logistika sohasiga oid masala Bellman-Ford algoritmidan foydalanib tahlil qilish va yechish ko'rib chiqildi.

Kalit so'zlar: Bellman-Ford algoritmi, og'irlikli graf, manfiy sikl, eng qisqa yo'l, logistika va yetkazib berish.

Abstract. This article provides information about the fields of application of the Bellman-Ford algorithm and how it is used in these fields. Analysis and solution of the problem related to the field of logistics using the Bellman-Ford algorithm was considered.

Keywords: Bellman-Ford algorithm, weighted graph, negative cycle, shortest path, logistics and delivery.

Аннотация. В статье представлена информация об областях применения алгоритма Беллмана-Форда и о том, как он используется в этих областях. Рассмотрен анализ и решение задачи, связанной с областью логистики с использованием алгоритма Беллмана-Форда.

Ключевые слова: алгоритм Беллмана-Форда, взвешенный граф, отрицательный цикл, кратчайший путь, логистика и доставка.

Bellman-Ford algoritmi og'irlikli grafda berilgan boshlang'ich uchda boshlanib boshqa uchga boorish uchun eng qisqa masofalarni toppish uchun ishlatiladigan algoritm hisoblanadi. Bellman-Ford algoritmi eng qisqa yo'lni hisoblash bilan bir qatorda manfiy og'irlikka ega bo'lgan sikllarni aniqlashga ham yordam beradi. Uning bu xususiyati ko'plab sohalarda foydalanishga imkon yaratadi. Quyida Bellman-Ford algoritmi eng ko'p qo'llaniladigan sohalar haqida qisqacha ma'lumot keltirilgan.

1. Telekommunikatsiya va tarmoqlarda.

- Routerlar uchun marshrutlash protokollari: Masalan RIP (Routing

Information Protocol) algoritmi Bellman- Ford algoritmiga asoslanadi. Bu protokol yo'llarni tahlil qilib, eng qisqa marshrutni aniqlab beradi.

2. Kompyuter ilmlari (Computer Science) va graflar nazariyasi.

- Yakka manbadan eng qisqa yo'llarni topish: Bellman-Ford algoritmi grafdagi ma'lum bir manbadan barcha tugunlarga eng qisqa yo'lni topishga yordam beradi.
- Manfiy sikllarni aniqlash: Bellman-Ford algoritmi sikl va yo'llar mavjud bo'lgan graflarda manfiy sikllarni aniqlay olishi sababli, bu bog'liq bo'lgan muammolarni tahlil qilishda foydalaniladi.

3. Sun'iy intellekt va o'yinlar.

- Yo'l tanlash strategiyalari: Ayrim o'yinlarda zararli hududlar (manfiy qiymatli) mavjud bo'lishi mumkin. Shunday sharoitlarda Bellman-Ford yordamida zarari kam bo'lgan yo'llar topiladi.
- Mustahkamlashli o'yinlar: Bu algoritm agentlar tomonidan mukofot va jarimalarni hisobga olgan holda o'rganish jarayonida qo'llaniladi. Harakatlar graf ko'rinishida foydalaniladi va eng qisqa yo'lni topish orqali maqsadga erishishda qo'llaniladi.

4. Ilmiy-texnik tadqiqotlar.

- Resurslarni optimal taqsimlash: Rivojlanayotgan mamlakatlar yoki korporatsiyalarning resurslarini (misol uchun, energiya, suv ta'minoti) minimal xarajat bilan taqsimlashda eng qisqa yo'l va eng kam xarajat masalalarini hal qilishda yordam beradi.
- Loyihalarni rejalashtirish va optimallashtirish: Turli loyihalarda resurslar va eng qisqa vaqt ichida maksimal samaradorlik bilan taqsimlash kerak bo'lganda foydalaniladi.

Logistika va transport tizimida Bellman-Ford algoritmidan foydalanish bir qancha qiziqarli masalalarni yechishda qo'l keladi. Quyida logistikaga oid masala keltirilgan.

Masala:

Kompaniya mahsulotlarini bir shahardan boshqasiga yetkazib berish uchun turli yo'nalishlarni ko'rib chiqmoqda. Har bir yo'nalishda yuk tashish xarajatlari mavjud lekin ba'zi yo'llarda chegirmalar yoki turli sabablarga ko'ra xarajatlar kamayishi mumkin. Bellman-Ford algoritmi yordamida kompaniya eng maqbul (arzon) yo'lni topishi mumkin.

- Grafdagi shaharlarga har birida yuk tashish xarajati berilgan qirralar bilan bog'langan tugunlar sifatida qaraladi.
- Bellman-Ford algoritmi barcha yo'nalishlarni tahlil qilib, boshlang'ich shahar (masalan, uni "A" deb belgilash mumkin) dan boshqa barcha shaharlarga eng kam xarajat sarflanadigan yo'lni topadi.

Algoritmning berilishi:

1. Har bir shaharni tugun sifatida, har bir yo'nalishni qirra sifatida tasvirlaydigan model yaratiladi.
2. Barcha qirralar tahlil qilinadi va tanlanga boshlang'ich tugundan ("A" shahardan) boshqa barcha tugunlarga eng qisqa yo'llar hisoblanadi.
3. Agar qaysidir takroriy bosqichda manfiy sikl aniqlansa, bu yo'nalishda chiqimni kamaytirishga imkon yaratadi.

Ushbu algoritmgga C# tilida dastur tuzildi:

```
using System;
```

```
using System.Collections.Generic;
```

```
namespace Bellman_Ford_Algorithm
```

```
{
```

```
class BellmanFord
```

```
{
```

```
public class Edge
```

```
{
```

```
public int Source, Destination, Weight;
```

```
public Edge(int source, int destination, int weight)
```

```
{
```

```
Source = source;
```

```
Destination = destination;
```

```
Weight = weight;
```

```
}
```

```
}
```

```
private int vertices;
```

```
private List<Edge> edges;
```

```
public BellmanFord(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 RunAlgorithm(int startVertex)
{
    int[] distances = new int[vertices];
    for (int i = 0; i < vertices; i++)
        distances[i] = int.MaxValue;

    distances[startVertex] = 0;

    for (int i = 1; i <= vertices - 1; i++)
    {
        foreach (Edge edge in edges)
        {
            int u = edge.Source;
            int v = edge.Destination;
            int weight = edge.Weight;

            if (distances[u] != int.MaxValue && distances[u] + weight < distances[v])
                distances[v] = distances[u] + weight;
        }
    }

    foreach (Edge edge in edges)
    {
        int u = edge.Source;
        int v = edge.Destination;
        int weight = edge.Weight;

        if (distances[u] != int.MaxValue && distances[u] + weight < distances[v])
        {
            Console.WriteLine("Grafda manfiy qiymatli sikl mavjud.");
            return;
        }
    }

    Print(distances, startVertex);
}

private void Print(int[] distances, int startVertex)
{
```

```

    Console.WriteLine("Boshlang'ich tugundan boshqa tugungacha bo'lgan
masofa " + startVertex + ":");
    for (int i = 0; i < vertices; i++)
    {
        if (distances[i] == int.MaxValue)
            Console.WriteLine(i + "tugunga " + "yetib bo'lmaydi.");
        else
            Console.WriteLine(i+"tugungacha bo'lgan masofa:" + distances[i]);
    }
}

class Program
{
    static void Main()
    {
        int vertices = 5;
        BellmanFord bf = new BellmanFord(vertices);

        bf.AddEdge(0, 1, -1);
        bf.AddEdge(0, 2, 4);
        bf.AddEdge(1, 2, 3);
        bf.AddEdge(1, 3, 2);
        bf.AddEdge(1, 4, 2);
        bf.AddEdge(3, 2, 5);
        bf.AddEdge(3, 1, 1);
        bf.AddEdge(4, 3, -3);

        bf.RunAlgorithm(0); // 0 tugundan boshlaydi
        Console.ReadKey();
    }
}

```

Yuqoridagi dasturda berilgan boshlang'ich tugundan (ya'ni 0 dan) boshqa barcha tugunlarga eng qisqa masofalar aniqlanadi hamda u grafda manfiy og'irlikka ega bo'lgan sikl mavjudligini ham tekshiradi.

Grafda tugunlar va qirralar quyidagicha berilgan:

Tugunlar: 0, 1, 2, 3, 4

Qirralar va ularning og'irliklari:

- 0 → 1 (og'irligi=-1)
- 0 → 2 (og'irligi=4)
- 1 → 2 (og'irligi=3)
- 1 → 3 (og'irligi=2)
- 1 → 4 (og'irligi=2)
- 2 → 2 (og'irligi=5)
- 3 → 1 (og'irligi=1)
- 4 → 3 (og'irligi=-3)

Xulosa

Bellman-Ford algoritmi og'irlik graf muammolarini hal qilishda, ayniqsa, manfiy og'irlikka ega sikllarni aniqlash imkoniyati bilan boshqa algoritmlardan ajralib turadi. Maqolada algoritmnining telekommunikatsiya, moliyaviy bozorlar, logistika va sun'iy intellekt kabi sohalarda qo'llanilishi batafsil yoritilgan. Ushbu algoritmnining amaliy masalalarda, xususan, transport xarajatlarini optimallashtirish va tarmoqlarni samarali boshqarishda qo'llanilishi uning ahamiyatini yanada oshiradi.

Amaliy dasturiy misol va algoritmnining ishlash prinsipi, ayniqsa, grafik modellashtirishni o'rganuvchilar uchun foydali bo'lib, bu mavzuning kengroq auditoriya uchun qiziqarli ekanligini ta'kidlaydi.

Foydalanilgan adabiyotlar:

1. Shimbel A. *Structure in communication nets*. New York: Polytechnic Press of the Polytechnic Institute of Brooklyn, 1955.
2. Marcin Jamro. *C# Data Structures and Algorithms*, Packt Publishing, 2018.
3. Cormen.T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. *Introduction to Algorithms*. MIT Press, 2019.
4. L.R.Ford, Jr.,D.R.Fulkerson. *Flows in Networks*, Princeton University Press, 1962.
5. McMillan, Michael. *Data Structures Algorithms using C#*, New York: Cambridge University Press, 2007.
6. Тюкачев Н. А. *C#. Алгоритмы и структуры данных*, Санкт-Петербург : Лань, 2021.
7. Marcin Jamro. *C# Data Structures and Algorithms*. Second Edition. Published by Packt Publishing Ltd., in Birmingham, UK. 2024. – 349 p.
8. Дж.Эриксон. *Алгоритмы*.: – М.: " ДМК Пресс ", 2023. – 528 с.
9. Hemant Jain. *Data Structures & Algorithms using Kotlin*. Second Edition. in India. 2022. – 572 p.
- 10.Н. А. Тюкачев, В. Г. Хлебостроев. *C#. Алгоритмы и структуры данных: учебное пособие для СПО*. – СПб.: Лань, 2021. – 232 с.
- 11..Mykel J. Kochenderfer. Tim A. Wheeler. *Algorithms for Optimization*. Published by The MIT Press., in London, England. 2019. – 500 p.
12. Рафгарден Тим. *Совершенный алгоритм. Графовые алгоритмы и структуры данных*. – СПб.: Питер, 2019. - 256 с.

13. Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э. Структуры данных и алгоритмы. – М.: Вильямс, 2018. – 400 с.
14. 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.
15. 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.
16. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. *Theoretical aspects in the formation of pedagogical sciences*, 2(22), 90-96
17. 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.
18. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 425-429.
19. Raxmonjonovich, F. S. (2023). Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 434-438.
20. 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.
21. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 475-481.
22. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 482-486.
23. 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.
24. 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.
25. Raxmonjonovich, F. S. (2023). C# tilida ArrayList bilan ishlashning afzalliklari. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 470-474.
26. 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>.