

FORD-BELMAN ALGORITMI

Farmonov Sherzodbek Raxmonjonovich

Fargʻona davlat universiteti amaliy matematika va informatika kafedrası

katta oʻqituvchisi

[*farmonovsh@gmail.com*](mailto:farmonovsh@gmail.com)

Abdurashidov Ozodbek Xurshidbek oʻgʻli

Fargʻona davlat universiteti talabasi

[*abdurashidovozodbek339@gmail.com*](mailto:abdurashidovozodbek339@gmail.com)

Annotatsiya: Ushbu maqolada Ford-Belman algoritmi (shuningdek, Bellman-Ford algoritmi deb ham ataladi) haqida batafsil maʼlumot beriladi. Algoritmning ishlash prinsipi, uning murakkablik darajasi va qoʻllanilish sohalari koʻrib chiqiladi. Shuningdek, uning boshqa qisqa masofa topish algoritmlaridan, xususan, Dijkstra algoritmidan asosiy farqlari tahlil qilinadi.

Kalit soʻzlar: Ford-Belman algoritmi, manfiy ogʻirlik, graf tahlili, eng qisqa yoʻl, murakkablik.

Annotation: This article provides detailed information about the Ford-Belman algorithm (also known as the Bellman-Ford algorithm). The principle of operation of the algorithm, its level of complexity and areas of application are considered. Also, its main differences from other short distance finding algorithms, in particular Dijkstra's algorithm, are analyzed.

Key words: Ford-Belman algorithm, negative weight, graph analysis, shortest path, complexity.

Аннотация: В этой статье представлена подробная информация об алгоритме Форда-Бельмана (также известном как алгоритм Беллмана-Форда). Рассмотрен принцип работы алгоритма, его уровень сложности и области применения. Также проанализированы его основные отличия от других алгоритмов поиска коротких расстояний, в частности от алгоритма Дейкстры

Ключевые слова Ключевые слова: алгоритм Форда-Бельмана, отрицательный вес, анализ графов, кратчайший путь, сложность.

Kirish. Ford-Belman algoritmi grafiklarda manfiy og'irlikli yo'llar mavjud bo'lgan holatlarda eng qisqa yo'lni topish uchun keng qo'llaniladi. Bu algoritmi manfiy tsikllarni aniqlash imkoniyatiga ega bo'lgan bir nechta algoritmlardan biridir. Ushbu algoritmi dastlab Richard Bellman va Lester Ford tomonidan mustaqil ravishda ishlab chiqilgan bo'lib, hozirgi kunda transport, tarmoq oqimlarini optimallashtirish va boshqa sohalarda qo'llaniladi.

Algoritmi ning ishlash prinsipi

Ford-Belman algoritmi og'irlikli yo'naltirilgan grafda berilgan manbadan barcha boshqa tugunlarga eng qisqa masofalarni topish uchun ishlatiladi. Algoritmi ning asosiy ishlash prinsipi iteratsiyalar orqali grafikning barcha qirralari bo'ylab masofalarni yangilashdan iborat. Algoritmi quyidagi bosqichlardan iborat:

1. **Boshlang'ich qiymatlar o'rnatilishi:** Manba tugundan barcha tugunlarga masofa cheksiz (∞), manba tuguniga masofa esa 0 qilib belgilanadi.
2. **Qirralarni iteratsiya qilish:** Har bir qirra bo'ylab masofalar yangilanadi. Masalan, agar u tugundan v tugungacha $w(u,v)$ og'irlikli qirra mavjud bo'lsa va $d(u)+w(u,v)<d(v)$ unda $d(v)=d(u)+w(u,v)$ yangilanadi.
3. **Manfiy tsiklni aniqlash:** Oxirgi iteratsiyadan so'ng masofalar yana yangilanayotgan bo'lsa, grafikda manfiy tsikl mavjud, deb xulosa qilinadi.

Algoritmi ning murakkabligi

Ford-Belman algoritmi ning ishlash murakkabligi:

- **Vaqt murakkabligi:** $O(V \times E)$, bu yerda V tugunlar soni va E qirralar soni.
- **Xotira murakkabligi:** $O(V)$, chunki faqat tugunlarning masofalari va qirralar ma'lumotlari saqlanadi.

Ford-Belman va Dijkstra algoritmlari taqqoslanishi

Ford-Belman algoritmi va Dijkstra algoritmi har ikkisi eng qisqa yo‘lni topish uchun ishlatiladi, lekin ular o‘zaro farqlanadi:

1. **Manfiy og‘irliklar:** Ford-Belman manfiy og‘irliklar bilan ishlay oladi, Dijkstra esa bunday imkoniyatga ega emas.

2. **Ishlash murakkabligi:** Dijkstra algoritmi $O((V+E)\log V)$ bo‘lgan tezroq vaqt murakkabligiga ega, lekin manfiy og‘irliklar yo‘q holatlarda qo‘llaniladi.

3. **Iteratsiyalar soni:** Ford-Belman tugunlar soni $V-1$ marta iteratsiya qiladi, Dijkstra esa prioritet navbat asosida ishlaydi.

Algoritmining qo‘llanilish sohalari

- **Transport tizimlari:** Shahar transport tizimlarida eng qisqa yo‘lni aniqlash.
- **Kompyuter tarmoqlari:** Paketlar uzatilishini optimallashtirish.
- **Iqtisodiy modellar:** Resurslar taqsimotini optimallashtirish.
- **Ilmiy tadqiqotlar:** Tarmoq grafika modellarida analitik tadqiqotlar o‘tkazish.

Ford-Belman algoritmini C# dasturlash tilida ifodalanishi.

```

using System;
using System.Collections.Generic;
class FordBellman
{ // Graf qirralarini ifodalovchi sinf public class Edge
{ public int Source, Destination, Weight;
public Edge(int source, int destination, int weight)
{Source = source;Destination = destination;
Weight = weight;}}
public static void FordBellmanAlgorithm(List<Edge> edges, int
vertices, int source)
{ // Masofalarni saqlash uchun massiv int[] distance = new
int[vertices]; for (int i = 0; i < vertices; i++)distance[i] = int.MaxValue;

```

```

distance[source] = 0;// Qirralar bo'ylab masofalarni yangilash for (int i = 0;
i < vertices - 1; i++){foreach (var edge in edges){if (distance[edge.Source] !=
int.MaxValue && distance[edge.Source] + edge.Weight <
distance[edge.Destination]){ distance[edge.Destination] =
distance[edge.Source] + edge.Weight;}}}
// Manfiy tsikllarni tekshirish
foreach (var edge in edges)
{if (distance[edge.Source] != int.MaxValue && distance[edge.Source]
+ edge.Weight < distance[edge.Destination])
{Console.WriteLine("Grafda manfiy tsikl mavjud!");return;}}
// Natijalarni chop etish
Console.WriteLine("Manba tugundan eng qisqa masofalar:");
for (int i = 0; i < vertices; i++)
{Console.WriteLine($"Tugun {i}: {(distance[i] == int.MaxValue ? "∞" :
distance[i].ToString())}");}}static void Main(string[] args){// Tugunlar va
qirralar soni int vertices = 5;
// Qirralarni ro'yxati
List<Edge> edges = new List<Edge>
{new Edge(0, 1, 4),
new Edge(0, 2, 5),
new Edge(1, 3, -3),
new Edge(2, 3, 3),
new Edge(3, 4, 2)
};// Manba tugunint source = 0;
// Algoritmni ishga tushirish
FordBellmanAlgorithm(edges, vertices, source);}
Kodni tushuntirish
1. Edge sinfi:

```

2. Graf qirralarini saqlash uchun alohida sinf yaratilgan. Har bir qirra boshlang'ich tugun (Source), oxirgi tugun (Destination), va qirra og'irligi (Weight) qiymatlarini o'z ichiga oladi.

3. **Masofalarni hisoblash:**

4. distance massivida manba tugundan har bir tugungacha bo'lgan eng qisqa masofalar saqlanadi. Boshlang'ichda barcha masofalar int.MaxValue (∞) qilib o'rnatiladi, faqat manba tugun uchun masofa 0.

5. **Iteratsiya:**

Qirralar $V-1V-1V-1$ marta iteratsiya qilinib, har bir tugun uchun masofalar yangilanadi.

6. **Manfiy tsiklni tekshirish:**

7. Algoritm oxirida qirralar yana bir marta tekshiriladi. Agar masofalar o'zgarayotgan bo'lsa, grafikda manfiy tsikl mavjud deb xulosa qilinadi.

8. **Natijalarni chiqarish:** Har bir tugun uchun eng qisqa masofalar konsolda chop etiladi.

Xulosa:

Ford-Belman algoritmi eng qisqa yo'l masalalarini yechishda universal usul hisoblanadi. U manfiy og'irlikli grafiklarda ishlay oladi, lekin nisbatan sekinroq ishlash murakkabligiga ega. Dasturlashda Ford-Belman algoritmi o'zining oddiyligi va samaradorligi bilan katta ahamiyatga ega bo'lib, ko'plab sohalarda muvaffaqiyatli qo'llaniladi.

ADABIYOTLAR:

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009).
Introduction to Algorithms (3rd Edition).
MIT Press.

Sedgewick, R., & Wayne, K. (2011).
Algorithms (4th Edition).
Addison-Wesley.

Kleinberg, J., & Tardos, É. (2005).

Algorithm Design.

Pearson.

Dasgupta, S., Papadimitriou, C., & Vazirani, U. (2006).

Algorithms.

McGraw-Hill.

Bellman, R. (1958).

"On a routing problem."

Quarterly of Applied Mathematics, 16(1), 87–90.

Ford, L. R. (1956).

"Network flow theory."

RAND Corporation Research Memorandum.

Moore, E. F. (1957).

"The shortest path through a maze."

Proceedings of an International Symposium on the Theory of Switching.

Cherkassky, B. V., & Goldberg, A. V. (1996).

"Negative-cycle detection algorithms."