

## **JONSON ALGORITMI VA UNING HOZIRGI TEXNOLOGIYALARDAGI O'RNI**

**Farmonov Sherzodbek Raxmonjonovich**

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

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

**Ergashboyev Muhammadrasul Ibrohim o'g'li**

*Farg'ona davlat universiteti talabasi*

*e-mail: [muhammadrasulergashboyev472@gmail.com](mailto:muhammadrasulergashboyev472@gmail.com)*

**Anotatsiya.** Ushbu maqolada Jonson algoritmining nazariy asosi, ishlash prinsipi va uning graflar nazariyasidagi o'rni tahlil qilinadi. Jonson algoritmi salbiy og'irliklarga ega grafdagi har bir tugun jufti orasidagi eng qisqa yo'llarni samarali hisoblash imkonini beradi. Maqolada algoritmning asosiy bosqichlari, ya'ni Bellman-Ford va Deykstra algoritmlariga asoslangan ishlash jarayoni, qo'llaniladigan ma'lumotlar tuzilmalari va hisoblash murakkabligi haqida batafsil ma'lumot beriladi.

**Kalit so'zlar:** Graf, Jonson algoritmi, Deykstra algoritmi, Floyd-Warshall, logistika, og'irlik, eng qisqa yo'l, salbiy og'irlik, tugun, yo'naltirilgan graf, Big data, optimallashtirish, Bellman-Ford algoritmi, sun'iy intellekt

**Annotation.** This article analyzes the theoretical basis of Johnson's algorithm, the principle of operation and its role in graph theory. Johnson's algorithm allows you to efficiently calculate the shortest paths between each pair of nodes in a graph with negative weights. The article provides detailed information about the main steps of the algorithm, namely, the operation process based on the Bellman-Ford and Dijkstra algorithms, the data structures used, and the computational complexity.

**Keywords:** Graph, Johnson algorithm, Dijkstra algorithm, Floyd-Warshall, logistics, weight, shortest path, negative weight, node, directed graph, Big data, optimization, Bellman-Ford algorithm, artificial intelligence

**Аннотация.** В данной статье анализируются теоретические основы алгоритма Джонсона, принцип работы и его роль в теории графов. Алгоритм Джонсона позволяет эффективно рассчитывать кратчайшие пути между каждой парой узлов графа с отрицательными весами. В статье представлена подробная информация об основных шагах алгоритма, а именно о процессе работы на основе алгоритмов Беллмана-Форда и Дейкстры, используемых структурах данных и вычислительной сложности.

**Ключевые слова:** Граф, алгоритм Джонсона, алгоритм Дейкстры,

Флойда-Уоршалла, логистика, вес, кратчайший путь, отрицательный вес, узел, ориентированный граф, большие данные, оптимизация, алгоритм Беллмана-Форда, искусственный интеллект.

### **Kirish**

Graf nazariyasi zamonaviy hisoblash tizimlarida keng qo'llaniladi. U yo'llarni optimallashtirish, logistika, tarmoq topologiyasini tahlil qilish va ko'plab boshqa masalalarni yechishda muhim ahamiyatga ega. Jonson algoritmi (Johnson's Algorithm) — grafda har bir juftlik orasidagi eng qisqa yo'lni topishga mo'ljallangan kuchli algoritm. U ayniqsa, salbiy og'irlikli qirralarga ega grafalarni tahlil qilishda o'zining samaradorligi bilan ajralib turadi. Ushbu algoritm boshqa mashhur algoritmlar, masalan, Floyd–Warshall yoki Deykstra algoritmlari bilan solishtirilganda ko'pincha katta grafalar uchun ustunlikka ega.

### **Graflarga kirish.**

Graf nazariyasi zamonaviy hisoblash tizimlarida keng qo'llaniladi. U yo'llarni optimallashtirish, logistika, tarmoq topologiyasini tahlil qilish va ko'plab boshqa masalalarni yechishda muhim ahamiyatga ega. Jonson algoritmi (Johnson's Algorithm) — grafda har bir juftlik orasidagi eng qisqa yo'lni topishga mo'ljallangan kuchli algoritm. U ayniqsa, salbiy og'irlikli qirralarga ega grafalarni tahlil qilishda o'zining samaradorligi bilan ajralib turadi.

### **Graf haqida tushuncha**

Graf — bu tugunlar (vertices) va qirralardan (edges) tashkil topgan matematik obyekt. Grafik yordamida ko'plab real hayotdagi masalalar model qilinadi, masalan, yo'llar tarmog'i, ijtimoiy tarmoqlar, ma'lumotlar oqimi va boshqalar. Grafning asosiy elementlari quyidagilar:

#### **Tugunlar:**

Tugunlar — grafikning asosiy nuqtalari. Ular odatda ma'lum bir ob'yektlarni (shaharlar, odamlar, kompyuterlar) ifodalaydi.

#### **Qirralar:**

Qirralar — tugunlarni bog'lovchi yo'llar. Qirralar ikki turga bo'linadi:

**Yo'naltirilgan graf (Directed graph):** Qirralar yo'nalishga ega ( $A \rightarrow B$ ). Masalan, bir yo'nalishli ko'cha.

**Yo'naltirilmagan graf (Undirected graph):** Qirralar yo'nalishga ega emas ( $A \leftrightarrow B$ ). Masalan, ikki tomonlama yo'l.

#### **Og'irliklar:**

Grafikdagi qirralarga og'irliklar biriktiriladi, ular qirra bo'ylab o'tishning qiymati, masofasi yoki boshqa metrikani ifodalaydi. Masalan, yo'nalishning uzunligi yoki narxi.

### **Jonson algoritmi haqida**

Jonson algoritmi yo'llarni optimallashtirish muammosini hal qilishda foydalaniladi. Algoritmning asosiy g'oyasi grafning qirra og'irliklarini o'zgartirish orqali salbiy og'irlikli qirralarni ijobiyga aylantirishdir. Bu algoritm Bellman-Ford va Deykstra algoritmlarining kombinatsiyasiga asoslangan. Quyida algoritmning bosqichlari keltirilgan:

#### **Jonson algoritmining ishlash jarayoni**

##### **1. Grafga yangi tugun qo'shish**

Grafga sun'iy yangi tugun  $s$  qo'shiladi va u boshqa har bir tugunga og'irligi 0 bo'lgan qirralar bilan bog'lanadi. Bu bosqich quyidagilarni ta'minlaydi:

Bellman-Ford algoritmini qo'llash uchun barcha tugunlarni bog'laydigan boshlang'ich nuqta yaratadi.

Misol: Agar grafikda  $V$  tugun bo'lsa, yangi tugun qo'shilgandan so'ng  $V+1$  tugunli bo'ladi.

##### **2. Bellman-Ford algoritmini qo'llash**

Yangi qo'shilgan tugunni boshlang'ich tugun deb olib, Bellman-Ford algoritmini bajariladi. U quyidagilarni ta'minlaydi:

Har bir tugunga eng qisqa masofalarni hisoblaydi.

Salbiy og'irlikli sikl mavjudligini tekshiradi.

Natija: Bellman-Ford algoritmi har bir tugun uchun masofa qiymatlari  $h(u)$  qaytaradi:

$h(u)$ : Tugun  $u$  uchun minimal masofa.

Muhim: Agar grafda salbiy og'irlikli sikl bo'lsa, Jonson algoritmi ishlamaydi va "Salbiy sikl mavjud" xatosi qaytariladi.

##### **3. Qirralar og'irliklarini o'zgartirish (Reweighting)**

Bellman-Ford algoritmidan olingan qiymatlar  $h(u)$  yordamida grafning barcha qirralari og'irligi quyidagicha qayta hisoblanadi:

$$w'(u,v) = w(u,v) + h(u) - h(v)$$

bu yerda:

$w(u,v)$  — qirraning dastlabki og'irligi,

$h(u)$  va  $h(v)$  — Bellman-Ford natijasi bo'yicha  $u$  va  $v$  tugunlarning qiymatlari.

Nima uchun bu kerak?

Qirralarning og'irligini o'zgartirish orqali barcha salbiy og'irlikli qirralar ijobiy bo'ladi.

Eng qisqa yo'llar tahliliga xalaqit qiluvchi salbiy qiymatlar bartaraf etiladi.

##### **4. Deykstra algoritmini qo'llash**

Qayta hisoblangan graf uchun har bir tugunni boshlang'ich nuqta sifatida olib, Deykstra algoritmi yordamida boshqa barcha tugunlarga eng qisqa masofalar hisoblanadi.

Bu bosqichning maqsadi:

Har bir tugundan boshqa tugunlarga eng qisqa yo'llarni topish.

Har bir tugundan V marta Deykstra algoritmini qo'llash orqali eng qisqa yo'llar matritsasi ( $d[u][v]$ ) hosil bo'ladi.

Masofalarni qayta tiklash:

Qayta hisoblangan masofalarni ( $w'(u, v)$ ) asl og'irlik qiymatlariga quyidagicha qaytariladi:

$$d(u, v) = w'(u, v) - h(u) + h(v)$$

Algoritmning yakuniy natijasi

Jonson algoritmi natijasida har bir tugun jufti orasidagi eng qisqa masofa qiymatlari qaytariladi.

### **Ishlash murakkabligi**

Jonson algoritmining hisoblash murakkabligi:

$$O(V \cdot E + V \cdot V \cdot \log V)$$

bu yerda:

V — tugunlar soni,

E — qirralar soni.

Bellman-Ford  $O(V \cdot E)$  vaqt oladi, Deykstra esa  $O(V \cdot \log V + E)$  vaqt oladi va bu V marta bajariladi.

### **Afzalliklari**

**Moslashuvchanlik:** Salbiy og'irlikli qirralarga ega grafalar bilan ishlay oladi.

**Tejamkorlik:** Grafning zichligiga qarab Floyd-Warshall algoritmidan samaraliroq.

**Kombinatsion yondashuv:** Bellman-Ford va Deykstra algoritmlarining afzalliklarini birlashtiradi.

### **Hozirgi texnologiyalardagi o'rni**

#### **Yo'llarni optimallashtirish:**

Transport tizimlari, masalan, Google Maps va Yandex.Navigator kabi xizmatlar yo'nalishlarni rejalashtirishda Jonson algoritmini qisman yoki to'liq qo'llaydi.

#### **Tarmoq topologiyasi:**

Kompyuter tarmoqlarida ma'lumot uzatish marshrutlarini optimallashtirishda qo'llaniladi.

#### **Big Data va ma'lumotlarni tahlil qilish:**

Katta grafik ma'lumotlarini tahlil qilish, ijtimoiy tarmoqlarni tahlil qilishda ham Jonson algoritmi foydalaniladi.

#### **AI va robototexnika:**

Jonson algoritmi sun'iy intellekt tizimlarida marshrutlarni belgilash va ob'ektlarni kuzatish uchun ishlatiladi.

**Masala:**

Berilgan yo'llar grafigi quyidagicha bo'lsin:

Tugunlar:                    A,                    B,                    C,                    D

Yo'llar va og'irliklari:

- A -> B = 4
- A -> C = 1
- C -> B = 2
- B -> D = 1
- C -> D = 5

Grafni matritsa shaklida tasvirlaymiz:

A B C D  
A [0, 4, 1, inf]  
B [inf, 0, inf, 1]  
C [inf, 2, 0, 5]  
D [inf, inf, inf, 0]

**Vazifa:**

Berilgan grafikda barcha juftliklar orasidagi eng qisqa yo'lni toping. Ya'ni, A dan D ga yoki C dan A ga bo'lgan eng qisqa yo'lni hisoblang.

**C# Kodining Yechimi:**

Jonson algoritmi ko'pincha Deykstra algoritmi yordamida ishlaydi. Avval grafikni o'zgartiramiz, so'ngra har bir tugundan Deykstra algoritmi yordamida eng qisqa masofalarni hisoblaymiz.

**C# kod:**

```
using System;
class JohnsonAlgorithm
{
    const int INF = int.MaxValue;
    // Deykstra algoritmi (Eng qisqa yo'lni topish)
    public static void Dijkstra(int[,] graph, int start, int n, int[] dist)
    {
        bool[] sptSet = new bool[n];
        dist[start] = 0;

        for (int count = 0; count < n - 1; count++)
        {
            int u = MinDistance(dist, sptSet, n);
```

```

        sptSet[u] = true;
        for (int v = 0; v < n; v++)
        {
            if (!sptSet[v] && graph[u, v] != INF && dist[u] != INF && dist[u] +
graph[u, v] < dist[v])
            {
                dist[v] = dist[u] + graph[u, v];
            }
        }
    }
}
// Eng kichik masofani topish
public static int MinDistance(int[] dist, bool[] sptSet, int n)
{
    int min = INF, minIndex = -1;
    for (int v = 0; v < n; v++)
    {
        if (!sptSet[v] && dist[v] <= min)
        {
            min = dist[v];
            minIndex = v;
        }
    }
    return minIndex;
}

// Jonson algoritmi
public static void Johnson(int[,] graph, int n)
{
    // Deykstra algoritmi yordamida barcha juftliklar orasidagi eng qisqa
    // masofalarni hisoblash
    for (int i = 0; i < n; i++)
    {
        int[] dist = new int[n];
        for (int j = 0; j < n; j++) dist[j] = INF;
        Dijkstra(graph, i, n, dist);
    }
}

```

Console.WriteLine(\$"Tugun {i} dan barcha boshqa tugunlarga eng qisqa yo'llar:");

```
for (int j = 0; j < n; j++)
{
    if (dist[j] == INF)
        Console.WriteLine($"Tugun {i} dan {j} tugunga yo'l mavjud emas");
    else
        Console.WriteLine($"Tugun {i} dan {j} tugunga eng qisqa yo'l:
{dist[j]}");
}
Console.WriteLine();
}
}

static void Main()
{
    // Grafikni matritsa ko'rinishida yaratamiz
    // 4 tugunli grafik (A=0, B=1, C=2, D=3)
    int[,] graph = new int[4, 4]
    {
        { 0, 4, 1, INF },
        { INF, 0, INF, 1 },
        { INF, 2, 0, 5 },
        { INF, INF, INF, 0 }
    };

    int n = 4; // Tugunlar soni
    Console.WriteLine("Jonson algoritmi natijalari:");
    Johnson(graph, n);
}
}
```

#### Kodning ishlash tartibi:

1. **Deykstra algoritmi:** Har bir tugundan (A, B, C, D) boshqa barcha tugunlarga eng qisqa yo'lni hisoblaydi.
2. **Jonson algoritmi:** Har bir tugundan Deykstra algoritmini chaqirib, barcha juftliklar orasidagi eng qisqa masofalarni hisoblaydi.

#### Kodning natijalari:

Masalan, quyidagi natijalar hosil bo'ladi:

Jonson algoritmi natijalari:

Tugun 0 dan barcha boshqa tugunlarga eng qisqa yo'llar:

Tugun 0 dan 0 tugunga eng qisqa yo'l: 0

Tugun 0 dan 1 tugunga eng qisqa yo'l: 4

Tugun 0 dan 2 tugunga eng qisqa yo'l: 1

Tugun 0 dan 3 tugunga eng qisqa yo'l: 5

Tugun 1 dan barcha boshqa tugunlarga eng qisqa yo'llar:

Tugun 1 dan 0 tugunga yo'l mavjud emas

Tugun 1 dan 1 tugunga eng qisqa yo'l: 0

Tugun 1 dan 2 tugunga yo'l mavjud emas

Tugun 1 dan 3 tugunga eng qisqa yo'l: 1

Tugun 2 dan barcha boshqa tugunlarga eng qisqa yo'llar:

Tugun 2 dan 0 tugunga yo'l mavjud emas

Tugun 2 dan 1 tugunga eng qisqa yo'l: 2

Tugun 2 dan 2 tugunga eng qisqa yo'l: 0

Tugun 2 dan 3 tugunga eng qisqa yo'l: 5

Tugun 3 dan barcha boshqa tugunlarga eng qisqa yo'llar:

Tugun 3 dan 0 tugunga yo'l mavjud emas

Tugun 3 dan 1 tugunga yo'l mavjud emas

Tugun 3 dan 2 tugunga yo'l mavjud emas

Tugun 3 dan 3 tugunga eng qisqa yo'l: 0

#### **Tavsiyalar:**

- Bu kod yordamida barcha juftliklar orasidagi eng qisqa yo'llarni hisoblash mumkin.
- Masala o'rganish uchun yaxshi boshlanish nuqtasidir va grafikka yangi qirralar qo'shib yanada murakkablashtirish mumkin.

#### **Foydalanilgan adabiyotlar:**

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. *Introduction to Algorithms*. MIT Press, 2009.
2. Sedgewick, R., & Wayne, K. *Algorithms*. Addison-Wesley, 2011.
3. Bondy, J. A., & Murty, U. S. R. *Graph Theory with Applications*. Elsevier, 1976.
4. Johnson, D. B. "A Faster Algorithm for All-Pairs Shortest Paths." *Journal of the ACM (JACM)*, 1977.
5. Tarjan, R. E., & Ramalingam, G. *Efficient Graph Algorithms for Shortest Paths*. SIAM Journal on Computing.
6. Skiena, S. S. *The Algorithm Design Manual*. Springer, 2008.
7. McMillan, M. *Data Structures and Algorithms in C#*. O'Reilly Media, 2007.



8. "Johnson Algorithm for All Pairs Shortest Paths," *GeeksforGeeks*: <https://www.geeksforgeeks.org>
9. "Johnson's Algorithm," *Wikipedia*: <https://en.wikipedia.org>
10. "Introduction to Graph Algorithms," *TutorialsPoint*:
  - a. <https://www.tutorialspoint.com>
11. 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.
12. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. *Theoretical aspects in the formation of pedagogical sciences*, 2(22), 90-96
13. 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.
14. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 425-429.
15. Raxmonjonovich, F. S. (2023). Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 434-438.
16. 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.
17. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 475-481.
18. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 482-486.
19. 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.
20. 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.
21. Raxmonjonovich, F. S. (2023). C# tilida ArrayList bilan ishlashning afzalliklari. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 470-474.

22. 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>.
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. Farmonov, S., & Rasuljonova, Z. (2024). OB'EKTGA YO'NALTIRILGAN DASTURLASH ZAMONAVIY DASTURLASHNING ASOSI SIFATIDA. *Центральноазиатский журнал образования и инноваций*, 3(1), 83-86.
25. 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.
26. 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.
27. Raxmonjonovich, F. S. (2024). C# VA MASHINA TILI. *Ta'lim innovatsiyasi va integratsiyasi*, 12(1), 59-62.
28. Farmonov, S. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. *Центральноазиатский журнал образования и инноваций*, 2(12 Part 2), 71-74.