

JONSON ALGORITMI BILAN BOG'LIQ MUAMMOLAR VA ULARNI YECHISH USULLARI

Farmonov Sherzodjon Raxmonjonovich

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

farmonovsh@gmail.com

Abdurasulova Gulsanam Nabijon qizi

*Farg'ona davlat universiteti Amaliy matematika
yo'nalishi 2-kurs talabasi*

gulsanamabdurasulova978@gmail.com

Annotatsiya. Ushbu maqolada Jonson algoritmi bilan bog'liq yuzaga keladigan asosiy muammolar va ularni yechish usullari tahlil qilingan. Maqolada algoritmning kompyuter resurslarini ko'p iste'mol qilishi, noaniq va murakkab holatlar; vaqt murakkabligi, xatoliklarni aniqlash hamda ma'lumotlar hajmi bilan bog'liq cheklovlar kabi asosiy muammolar ko'rib chiqilgan.

Kalit so'zlar: Jonson algoritmi, eng qisqa yo'l masalasi, algoritmni optimallashtirish, kompyuter resurslar, parallel hisoblash, vaqt murakkabligi, xatoliklarni aniqlash.

Abstract. This article analyzes the main problems associated with Johnson's algorithm and their solutions. The article considers the main problems of the algorithm, such as high consumption of computer resources, ambiguous and complex cases, time complexity, error detection, and data size limitations.

Keywords: Johnson's algorithm, shortest path problem, algorithm optimization, computer resources, parallel computing, time complexity, error detection.

Абстрактный. В данной статье анализируются основные проблемы, связанные с алгоритмом Джонсона, и способы их решения. В статье рассмотрены основные проблемы алгоритма, такие как высокое

потребление компьютерных ресурсов, неопределенные и сложные ситуации, временная сложность, обнаружение ошибок и ограничения по объему данных.

***Ключевые слова:** алгоритм Джонсона, задача о кратчайшем пути, оптимизация алгоритма, ресурсы компьютера, параллельные вычисления, временная сложность, обнаружение ошибок.*

Kirish:

Jonson algoritmi– bu ikki bosqichli tizimlarda eng qisqa yo‘llarni topish uchun ishlatiladigan samarali algoritmdir. U asosan, muammolarni tez va samarali yechish maqsadida ishlab chiqilgan bo‘lib, ko‘plab sohalarda, jumladan, logistika, ishlab chiqarish va tarmoq tahlilida keng qo‘llaniladi. Algoritmning asosiy afzalligi shundaki, u barcha juft tugunlar orasidagi eng qisqa yo‘lni topish imkonini beradi va o‘rganishning samaradorligini sezilarli darajada oshiradi. Biroq, amaliy qo‘llanish jarayonida Jonson algoritmi bir qator muammolarni keltirib chiqarishi mumkin. Bu muammolar orasida kompyuter resurslarini ko‘p iste‘mol qilish, vaqt murakkabligi, noaniqliklar va murakkab holatlar, shuningdek, xatoliklarni aniqlash va ularni tuzatish kabi masalalar mavjud. Ushbu maqolada Jonson algoritmi bilan bog‘liq eng muhim muammolar tahlil qilinadi va ularni samarali yechish uchun qo‘llaniladigan usullar ko‘rib chiqiladi

Muhokama va natijalar

Jonson algoritmi, ikkita bosqichli tizimlarda eng qisqa yo‘llarni topish uchun samarali vosita bo‘lishiga qaramay, uning amaliy qo‘llanilishida bir qator muammolar paydo bo‘ladi. Maqolada keltirilgan muammolar va ularni yechish usullarini tahlil qilish orqali, algoritmni yanada samarali ishlatish uchun bir nechta muhim yondashuvlar taklif qilinadi.

Birinchi, kompyuter resurslarini ko‘p iste‘mol qilish masalasi, ayniqsa, katta hajmdagi graf va tarmoq tizimlari bilan ishlashda dolzarbdir. Jonson algoritmi, o‘zining ikki bosqichli tuzilmasi bilan, juda ko‘p hisoblash resurslarini talab qilishi mumkin. Bu holatda, parallel hisoblash va qisman

natijalardan foydalanish usullari samarali yechim bo'lishi mumkin. Parallel hisoblashni qo'llash orqali, algoritmning turli qismlari bir vaqtning o'zida ishlatilishi, umumiy hisoblash vaqtini sezilarli darajada qisqartirishi mumkin. Shuningdek, ba'zi qismlarini soddalashtirish va hisoblashlarni minimallashtirish orqali ham resurslarni tejash mumkin.

Ikkinchidan, noaniqliklar va murakkab holatlar masalasi graf tuzilmasining xususiyatlaridan kelib chiqadi. Graflarda qaytariladigan tugunlar yoki o'zgarimas qirralar mavjud bo'lsa, bu algoritmning samaradorligini pasaytirishi yoki xatoliklarga olib kelishi mumkin. Bunday holatlar uchun grafni normalizatsiya qilish yoki qaytariladigan tugunlar va qirralarni oldindan aniqlash, algoritmni soddalashtirishi va samaradorligini oshirishi mumkin.

Vaqt murakkabligi ham Jonson algoritmining asosiy cheklovlaridan biridir. Katta hajmdagi tizimlarda, algoritmning ishlash vaqti keskin oshadi. Bu masala uchun yangi optimallashtirilgan algoritmlar va hechsiz yo'llardan foydalanish yondashuvlari tavsiya etiladi. Misol uchun, agar grafda o'zgarishlar minimal bo'lsa, faqat o'zgaragan qismlar bilan ishlash orqali vaqtni sezilarli darajada tejash mumkin.

Natijalar

Jonson algoritmining samaradorligini oshirish uchun bir nechta muhim yechimlar va yondashuvlar ishlab chiqiladi. Parallel hisoblash va qisman natijalardan foydalanish algoritmni tezlashtirishga yordam beradi. Grafni normalizatsiya qilish va qaytariladigan tugunlarni aniqlash esa algoritmni murakkab holatlardan xalos qiladi va uning samaradorligini oshiradi. Optimallashtirilgan algoritmlar va yangi yondashuvlar, masalan, hechsiz yo'llardan foydalanish, algoritmning ishlash vaqtini qisqartirishga yordam beradi.

Masala: Jonson algoritmi bilan bog'liq muammolarni yechish

Masala shartlari:

Bir kompaniya o'zi ishlab chiqarish jarayonida ikkita bosqichli tizimda eng qisqa vaqtni hisoblash uchun eng qisqa yo'lni topishga harakat qilmoqda. Tizimning birinchi bosqichida turli mashinalar ishlaydi, ikkinchi bosqichda esa

yakuniy mahsulotlar yig'iladi. Har bir bosqichda turli operatsiyalarni amalga oshirish uchun mashinalar o'rtasida bog'lanishlar mavjud. Kompaniya, tarmoq orqali o'tish uchun eng qisqa vaqtni hisoblashda Jonson algoritmini qo'llamoqda.

Biroq, kompaniya quyidagi muammolar bilan duch keladi:

1. **Resurslar talabining oshishi:** Kompaniya katta hajmdagi tizimni tahlil qilganda, Jonson algoritmi resurslarni ko'p talab qilishni boshlaydi. Bu esa tizimning samaradorligini pasaytiradi.

2. **Murakkab holatlar:** Grafikda ba'zi tugunlar va qirralar takrorlanadi, bu esa algoritmnning ishlashida noaniqliklarga olib keladi..

3. **Vaqt murakkabligi:** Katta hajmdagi tizimlarda, algoritmnning ish vaqtining oshishi vaqti bilan bog'liq muammolarni keltirib chiqaradi.

4. **Xatoliklarni aniqlash:** Algoritm ma'lumotlari to'g'ri bo'lmasligi sababli, natijalar ishonchsiz va noto'g'ri chiqadi.

Kutilgan natijalar:

- Eng qisqa yo'lni aniqlash va resurslarni tejash uchun samarali yondashuvlar.

- Takrorlanadigan tugunlar va qirralarni aniqlash va ularni bartaraf etish usullari.

- Algoritmning samaradorligini oshirish uchun optimallashtirish texnikalari.

- Xatoliklarni aniqlash va to'g'rilash uchun kerakli mexanizmlar.

. Jonson algoritmi eng qisqa yo'l masalasini yechishda ishlatiladi, lekin ikkita bosqichli tizimlarda ishlashga qaratilgan. Quyidagi masalada, tarmoqdagi barcha juft tugunlar orasida eng qisqa yo'lni hisoblash uchun Jonson algoritmi qanday ishlashini C# dasturida ko'rsatilgan. Bu yerda tarmoqni graflar yordamida tasvirlaydi va algoritmni kodlashda ba'zi optimallashtirishlar kiritiladi.

Masala: C# dasturilash tilida Jonson algoritmi bilan eng qisqa yo'llarni hisoblash dasturi tuzilsin.

```
using System;
```

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

```

public class Graph{ private int V; // Vertex (Tugunlar) soni
private List<Tuple<int, int, int>> edges; // Grafdagi qirralar: (manba
tugun, maqsad tugun, og'irlik)
// Konstruktor
public Graph(int v) { V = v; edges = new List<Tuple<int, int, int>>();}
// Qirra qo'shish: (manba tugun, maqsad tugun, og'irlik)
public void AddEdge(int u, int v, int weight)
edges.Add(new Tuple<int, int, int>(u, v, weight)); }
// Dijkstra algoritmi yordamida eng qisqa yo'llarni topish
private int[] Dijkstra(int src, int[] dist) { var pq = new
SortedSet<Tuple<int, int>>(); // Priority Queue (Min-Heap tarzida ishlaydi)
pq.Add(new Tuple<int, int>(0, src)); // Boshlang'ich manzildan
boshlash dist[src] = 0; // Boshlang'ich tugunni 0 ga tenglaymiz
while (pq.Count > 0 {
var current = pq.Min;
pq.Remove(current);
int u = current.Item2;
int currentDist = current.Item1;
// Agar bu yo'l avvalgi yo'ldan yaxshi bo'lsa
if (currentDist > dist[u])
continue;
// Qo'shni tugunlarga o'tish
foreach (var edge in edges)
{ int v = edge.Item2;
if (edge.Item1 == u)
{ int weight = edge.Item3;
if (dist[v] > dist[u] + weight)
{ dist[v] = dist[u] + weight pq.Add(new Tuple<int,
int>(dist[v], v)); } } }
return dist;
}

```

```

// Jonson algoritmi
public void JohnsonAlgorithm()
{ int[] dist = new int[V];
  for (int i = 0; i < V; i++)
    {dist[i] = int.MaxValue; // To'liq nomalum bo'lishi uchun katta
qiymat beramiz}

// Dijkstra algoritmi yordamida har bir tugun uchun eng qisqa yo'lni
toppish for (int i = 0; i < V; i++) {var result = Dijkstra(i, dist);
Console.WriteLine($"Eng qisqa yo'l ({i} dan boshqa tugunlarga: ");
for (int j = 0; j < V; j++)
{if (result[j] == int.MaxValue)
Console.WriteLine($" {i} dan {j} tugunga yo'l yo'q.");else
Console.WriteLine($" {i} dan {j} tugunga eng qisqa yo'l: {result[j]}");}}}}

public class Program
{ public static void Main(string[] args) {
// Grafikni yaratish (masalan, 5 ta tugun va turli qirralar bilan)
Graph graph = new Graph(5);
graph.AddEdge(0, 1, 10);
graph.AddEdge(0, 2, 3);
graph.AddEdge(1, 2, 1);
graph.AddEdge(1, 3, 2);
graph.AddEdge(2, 1, 4);
graph.AddEdge(2, 3, 8);
graph.AddEdge(3, 4, 7);
graph.AddEdge(4, 3, 9);

// Jonson algoritmi yordamida eng qisqa yo'llarni hisoblash
graph.JohnsonAlgorithm(); }

```

Dasturdagi Asosiy Qismlar:

1. **Graph klassi:**

- **AddEdge** metodi orqali grafdagi qirralarni qo‘shish. Qirralar (manba tugun, maqsad tugun, og‘irlik) tarzida saqlanadi.

- **Dijkstra** metodida eng qisqa yo‘lni hisoblash uchun Dijkstra algoritmi ishlatiladi. Bu metod grafdagi har bir tugundan boshlanib, eng qisqa yo‘lni topish uchun ishlaydi.

- **JohnsonAlgorithm** metodida Jonson algoritmi bajariladi, ya'ni har bir tugundan boshlab, barcha boshqa tugunlarga eng qisqa yo‘lni hisoblaydi.

2. Program klassi:

- Dasturda grafikni yaratish, unga qirralarni qo‘shish va Jonson algoritmini chaqirish uchun ishlatiladi.

- **graph.JohnsonAlgorithm()** orqali algoritm ishga tushiriladi va natijalar ekranda chiqariladi.

Dastur natijalari:

Eng qisqa yo'l (0 dan boshqa tugunlarga):

0 dan 0 tugunga eng qisqa yo'l: 0

0 dan 1 tugunga eng qisqa yo'l: 10

0 dan 2 tugunga eng qisqa yo'l: 3

0 dan 3 tugunga eng qisqa yo'l: 5

0 dan 4 tugunga eng qisqa yo'l: 12

Eng qisqa yo'l (1 dan boshqa tugunlarga):

1 dan 0 tugunga yo'l yo'q.

1 dan 1 tugunga eng qisqa yo'l: 0

1 dan 2 tugunga eng qisqa yo'l: 1

1 dan 3 tugunga eng qisqa yo'l: 2

1 dan 4 tugunga eng qisqa yo'l: 9

Xulosa qilib aytganda maqolada Jonson algoritmgiga doir asosiy masalalar va yechimlar tahlil qilindi. Resurslar talabining oshishi, murakkab holatlar, vaqt murakkabligi va xatoliklarni aniqlash kabi muammolar, dasturlashda va tizimlarni optimallashtirishda e'tiborli jihatlar hisoblanadi. C# da yozilgan dastur misolida, Jonson algoritmining eng qisqa yo‘l masalasini yechishda qanday ishlashi

ko'rsatildi. Dastur orqali, grafdagi barcha juft tugunlar orasidagi eng qisqa yo'lni hisoblash jarayoni va resurslarni optimallashtirishda qo'llanilishi mumkin bo'lgan yondashuvlar ko'rib chiqildi. Dijkstra algoritmi yordamida tugunlar o'rtasidagi eng qisqa yo'l aniqlandi va har bir tugundan boshqa barcha tugunlarga eng qisqa yo'l hisoblandi.

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* (T. 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'YEKTGA 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.