

IJTIMOYIY TARMOQLARDAGI BOG‘LANISHLARNI TAHLIL QILISHDA BFS ALGORITMINING AHAMIYATI

Sh.R.Farmonov

Farg‘ona davlat universiteti amaliy matematika va informatika kafedrasida o‘qituvchisi

farmonovsh@gmail.com

M.U.Jumaboyeva

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

Jumaboyevamohlaroy8@gmail.com

Annotatsiya. Maqola ijtimoiy tarmoqlardagi bog‘lanishlarni tahlil qilishda BFS algoritmining ishlash tamoyillari va qo‘llanilishini ko‘rib chiqadi. Algoritm yordamida foydalanuvchilar o‘rtasidagi eng qisqa bog‘lanish zanjiri topilishi misollar bilan tushuntiriladi. Shuningdek, uning afzalliklari va cheklovlari yoritiladi.

Kalit so‘zlar. BFS algoritmi, ijtimoiy tarmoq tahlili, eng qisqa bog‘lanish zanjiri, graf tahlili, kenglik bo‘yicha qidiruv, algoritmlar, tarmoq aloqalari, foydalanuvchilar bog‘lanishi.

Annotation. The article explores the application and working principles of the Breadth-First Search (BFS) algorithm in analyzing connections within social networks. It explains how the algorithm can be used to identify the shortest connection chain between users, supported by practical examples. Additionally, the advantages and limitations of BFS in social network analysis are highlighted.

Keywords. BFS algorithm, social network analysis, shortest connection chain, graph analysis, breadth-first search, algorithms, network connections, user connections.

Аннотация. В статье рассматриваются принципы работы и применение алгоритма поиска в ширину (BFS) для анализа связей в социальных сетях. Объясняется, как с помощью алгоритма можно определить кратчайшую цепочку связей между пользователями, приводятся практические примеры. Также освещаются преимущества и ограничения BFS в анализе социальных сетей.

Ключевые слова. Алгоритм BFS, анализ социальных сетей, кратчайшая цепочка связей, анализ графов, поиск в ширину, алгоритмы, сетевые связи, связи пользователей.

BFS algoritmi haqida qisqacha ma‘lumot. BFS algoritmi, asosan, graf yoki daraxt strukturasi bo‘yicha barcha tugunlarni qatlam-qatlam (yoki kenglik bo‘yicha) qidirish tamoyiliga asoslanadi. Bu algoritm, dastlab boshlang‘ich tugunni

tanlab, uning barcha qo'shni tugunlarini tekshirib, keyin esa keyingi qatlamdagi tugunlarni qidirish orqali ishlaydi. BFS algoritmi eng qisqa yo'lni topishda, tarmoq aloqalarini o'rganishda va ko'plab amaliy masalalarda qo'llaniladi. Uning asosiy afzalligi shundaki, har bir tugun faqat bir marta ko'rib chiqiladi va navbatda kutilayotgan tugunlar boshqalaridan avval tekshiriladi. Bu, uni ancha samarali qiladi. Aksincha, uning ba'zi cheklovlari ham mavjud, masalan, katta tarmoqlarda xotira iste'moli oshadi.

● **BFS algoritmining ishlash tamoyili.** BFS algoritmi graf yoki daraxtda tugunlarni qatlamma-qatlam qidirishga asoslanadi. Bu algoritmnining asosiy ishlash tamoyili - har bir tugunni bir marta ko'rib chiqish va tugunlarning barcha qo'shnilarini navbatga (queue) qo'shishdir. BFS algoritmi FIFO (First In, First Out) tamoyiliga asoslanadi, ya'ni birinchi navbatga qo'shilgan tugun birinchi bo'lib tekshiriladi.

Ishlash jarayoni:

→ Boshlang'ich tugunni tanlash: Algoritmida birinchi navbatga boshlang'ich tugun qo'shiladi.

→ Navbatni yaratish: Boshlang'ich tugun qo'shilgan navbatdan keyingi tugunlarni tekshirish uchun yangi tugunlar navbatga qo'shiladi.

→ Qo'shni tugunlarni tekshirish: Har bir tugunni tekshirib, uning qo'shnilarini navbatga qo'shamiz. Agar tugun ilgari tashrif buyurilmagan bo'lsa, uni navbatga qo'shish va ko'rish davom etadi.

→ Tugunlarni belgilash: Har bir tugun bir marta tashrif buyurilganini belgilab, tekshirilgan tugunlar ro'yxatiga qo'shiladi.

Misol uchun:

Aytaylik, grafda quyidagi tugunlar va bog'lanishlar mavjud:

A -- B -- C

| | |

D -- E -- F

1. Boshlang'ich tugun: A
2. Navbatga qo'shilgan tugunlar: B, D
3. Keyin navbatdagi tugunlar: C, E
4. Yakuniy tugunlar: F

BFS algoritmi shu tarzda ishlaydi: A ni tekshirib, B va D ni navbatga qo'shadi, keyin B ni tekshirib, C va E ni navbatga qo'shadi, va hokazo. Har bir tugun bir marta ko'rib chiqiladi.

● **BFS algoritmining afzalliklari va cheklovlari**

Afzalliklari:

→ Eng qisqa yo'lni topish: BFS algoritmi grafda ikki tugun orasidagi eng qisqa yo'lni topishda juda samarali. Chunki algoritm tugunlarni qatlamma-qatlam qidiradi va birinchi marta maqsadli tugunga yetganida, bu yo'l eng qisqa bo'ladi.

→ Oddiy va tushunarli ishlash prinsipi: BFSning ishlash tamoyili oddiy va intuitiv. Tugunlarni navbat bilan tekshirib borish, uning ishlashini tushunishni osonlashtiradi. Bu algoritmnin dasturlashda qo'llash va ishlab chiqish oson.

→ Hammasi bir marta ko'rib chiqiladi: BFS har bir tugunni faqat bir marta ko'rib chiqadi, bu esa algoritmnin samaradorligini oshiradi, chunki xotira va vaqt cheklovlari bo'yicha samarali ishlaydi.

→ Graphning barcha aloqalarini tahlil qilish: BFS ijtimoiy tarmoq tahlilida foydalanuvchilar o'rtasidagi barcha mumkin bo'lgan aloqalarni ko'rib chiqish imkoniyatini beradi. Shu tarzda tarmoqning to'liq strukturasi haqida tasavvur olish mumkin.

Cheklovlari:

→ Xotira iste'moli: BFS algoritmi katta grafiklarda ko'proq xotira talab qiladi, chunki u barcha tugunlarni navbatga qo'shadi va har bir tugun uchun tashrif buyurilganlar ro'yxatini saqlaydi. Bu holat graf kattalashgani sayin xotira iste'molini sezilarli darajada oshiradi.

→ Noqonuniy holatlar: Agar grafda sikl (qaytalanadigan yo'l) mavjud bo'lsa, BFS siklni qayta-qayta tekshirib chiqishi mumkin. Bu vaziyatda, ilgari tashrif buyurilgan tugunlarni belgilash kerak bo'ladi, aks holda algoritm oxiriga yetmasligi mumkin.

→ Yo'lni saqlashning yuqori xarajati: BFS, eng qisqa yo'lni topishda samarali bo'lsa-da, uni saqlash va qaytarish jarayoni ba'zi hollarda murakkab bo'lishi mumkin. Agar barcha mumkin bo'lgan yo'llarni saqlash kerak bo'lsa, bu jarayon ko'proq vaqt va xotira talab qiladi.

→ Keng tarmoqda sekin ishlash: BFS algoritmi o'zi tez ishlasa ham, keng va murakkab tarmoqlarda u sekin ishlashi mumkin. Katta graf strukturalarida tugunlar orasidagi aloqalarni tahlil qilishda vaqt sarfi oshadi.

• Ijtimoiy tarmoqdagi aloqalarni tahlil qilish

Ijtimoiy tarmoq — bu foydalanuvchilar va ularning o'zaro aloqalari (do'stlik, izdoshlik, yoki boshqa turdagi bog'lanishlar)dan iborat grafga o'xshash tizimdir. Bu grafidagi tugunlar foydalanuvchilarni, bog'lanishlar esa ularning o'zaro aloqalarini ifodalaydi. Ijtimoiy tarmoq tahlili, ayniqsa, foydalanuvchilar o'rtasidagi aloqalarni aniqlashda, BFS algoritmidan samarali foydalanish mumkin.

Masala.

Ijtimoiy tarmoqda ikki foydalanuvchi o'rtasidagi eng qisqa bog'lanish zanjirini topish kerak. BFS algoritmi yordamida bu masalani qanday yechish mumkin?

Grafning tuzilishi:

Tarmoqni graf shaklida tasvirlashimiz mumkin. Har bir foydalanuvchi — bu tugun, va ularning o'rtasidagi do'stliklar yoki aloqalar esa bog'lanishlar (qirralar)ni tashkil etadi.

Misol uchun:

A-- B-- C

| | |

D-- E—F

Bu yerda, A foydalanuvchisi B va D bilan do'st, B esa A, C va E bilan do'st. Agar biz A va F foydalanuvchilari o'rtasidagi eng qisqa yo'lni topmoqchi bo'lsak, BFS algoritmi yordamida ularni quyidagi tarzda qidiramiz:

1. Boshlang'ich tugun (A): A foydalanuvchisini tekshirib, uning qo'shnilarini (B va D) navbatga qo'shamiz.

2. Navbatdagi tugunlar (B va D): B foydalanuvchisiga o'tamiz va uning qo'shnilarini (C va E) tekshiramiz. D foydalanuvchisini ham tekshirib, uning qo'shnisini (E) ko'ramiz.

3. Keyingi tugunlar (C, E, F): C foydalanuvchisiga o'tamiz, E va F foydalanuvchilarini ko'ramiz. Nihoyat, F ga yetib kelamiz.

Bu jarayon orqali A va F orasidagi eng qisqa bog'lanish zanjiri: $A \rightarrow D \rightarrow E \rightarrow F$ bo'ladi.

Tarmoqda bog'lanishlarni tahlil qilish.

BFS yordamida ijtimoiy tarmoqdagi boshqa muhim masalalar ham yechilishi mumkin:

➔ Do'stlar zanjiri: BFS yordamida bir foydalanuvchi va uning barcha do'stlarini topish, yoki bir do'stning boshqa do'stlarini aniqlash.

➔ Tarmoqdagi eng yaqin foydalanuvchilar: Agar foydalanuvchi biror foydalanuvchiga eng yaqin do'stlarni topmoqchi bo'lsa, BFS algoritmi yordamida bu tahlilni o'tkazish mumkin.

BFS algoritmi ijtimoiy tarmoqda foydalanuvchilar o'rtasidagi aloqalarni o'rganish, ularning o'zaro bog'lanishlarini va ijtimoiy ta'sirni tahlil qilishda juda foydali bo'ladi.

Kod namunasi.

Endi BFS algoritmini C# tilida qanday yozish mumkinligi ko'rib chiqilsin. Ijtimoiy tarmoqdagi foydalanuvchilarni va ularning o'rtasidagi aloqalarni graf sifatida tasvirlanadi. Maqsad shundan iboratki, BFS algoritmi yordamida ikki foydalanuvchi orasidagi eng qisqa bog'lanish zanjirini topish.

```
class Graph
```

```
{
```

```
    // Grafning tugunlarini saqlash uchun Dictionary
```

```
    private Dictionary<string, List<string>> adjList;
```

```
    // Konstruktor - Grafni yaratish
```

```
    public Graph()
```

```
    {
```

```
        adjList = new Dictionary<string, List<string>>();
```

```

}
// Yangi bog‘lanishni qo‘shish
public void AddEdge(string node1, string node2)
{
    if (!adjList.ContainsKey(node1))
        adjList[node1] = new List<string>();
    if (!adjList.ContainsKey(node2))
        adjList[node2] = new List<string>();
    adjList[node1].Add(node2);
    adjList[node2].Add(node1); // ikki tomonlama bog‘lanish
}
// BFS yordamida eng qisqa yo‘lni topish
public List<string> BFS(string start, string goal)
{
    // Navbat va tashrif buyurilgan tugunlar ro‘yxatini yaratish
    Queue<string> queue = new Queue<string>();
    Dictionary<string, string> prevNodes = new Dictionary<string, string>();
    HashSet<string> visited = new HashSet<string>();
    queue.Enqueue(start);
    visited.Add(start);
    while (queue.Count > 0)
    {
        string current = queue.Dequeue();
        if (current == goal)
            return GetPath(prevNodes, goal);
        foreach (var neighbor in adjList[current])
        {
            if (!visited.Contains(neighbor))
            {
                visited.Add(neighbor);
                queue.Enqueue(neighbor);
                prevNodes[neighbor] = current;
            }
        }
    }
    return null; // Agar yo‘l topilmasa
}
// Eng qisqa yo‘lni qaytarish
private List<string> GetPath(Dictionary<string, string> prevNodes, string
goal)

```

```
{
    List<string> path = new List<string>();
    string current = goal;
    while (current != null)
    {
        path.Add(current);
        prevNodes.TryGetValue(current, out current);
    }
    path.Reverse();
    return path;
}
}
internal class Program
{
    public static void Main(string[] args)
    {
        // Grafni yaratish
        Graph g = new Graph();
        g.AddEdge("A", "B");
        g.AddEdge("A", "D");
        g.AddEdge("B", "C");
        g.AddEdge("B", "E");
        g.AddEdge("C", "F");
        g.AddEdge("D", "E");
        g.AddEdge("E", "F");
        // BFS yordamida eng qisqa yo'lni topish
        List<string> path = g.BFS("A", "F");
        if (path != null)
        {
            Console.WriteLine("Eng qisqa yo'l:");
            foreach (var node in path)
            {
                Console.Write(node + " ");
            }
        }
        else
        {
            Console.WriteLine("Yo'l topilmadi.");
        }
    }
}
```

}

Kodning ishlash prinsipi:

1. Graf tuzish: Graph sinfi yordamida ijtimoiy tarmoqdagi foydalanuvchilar va ularning o'rtasidagi aloqalar (do'stliklar) graf sifatida saqlanadi. AddEdge metodi orqali yangi bog'lanishlar qo'shiladi.

2. BFS algoritmi: BFS metodi orqali boshlang'ich foydalanuvchi (start) dan maqsadli foydalanuvchiga (goal) boradigan eng qisqa yo'l topiladi.

3. Eng qisqa yo'lni qaytarish: GetPath yordamida eng qisqa yo'lni qaytarib, foydalanuvchi tomonidan tanlangan ikki foydalanuvchi orasidagi aloqalarni aniqlaydi.

Agar "A" va "F" orasidagi eng qisqa bog'lanish zanjirini topish dasturida quyidagi natija chiqariladi:

Eng qisqa yo'l: A D E F

BFS algoritmi ijtimoiy tarmoq tahlilida foydalanuvchilar o'rtasidagi aloqalarni aniqlash va tarmoqdagi eng qisqa yo'llarni topishda juda foydalidir. Ushbu algoritmning asosiy afzalligi uning sodda va intuitiv ishlash tamoyilida yotadi, bu esa uni dasturlashda va tushunishda osonlashtiradi. BFS algoritmi foydalanuvchilar o'rtasidagi bog'lanishlar tarmog'ini kenglik bo'yicha tekshirish imkonini beradi, bu esa ijtimoiy tarmoqdagi aloqalar strukturasi chuqurroq tahlil qilishga yordam beradi. Shu bilan birga, BFS algoritmning ba'zi cheklovlari ham mavjud. Xususan, katta tarmoqlarda xotira iste'moli ortadi, va algoritmning ishlash tezligi ba'zan sekinlashishi mumkin. Shunday bo'lsa-da, ijtimoiy tarmoqdagi aloqalar, do'stlar zanjirlari va boshqa tahlil masalalarida BFS hali ham samarali va keng qo'llaniladigan usuldir. Umuman olganda, BFS algoritmi ijtimoiy tarmoq tahlilida, foydalanuvchi aloqalarini o'rganish va tarmoqdagi eng qisqa bog'lanishlarni aniqlashda juda muhim vosita bo'lib qoladi. Bu algoritmning sodda ishlash prinsipi va aniq natijalar berishi, uni yanada ommalashtiradi va amaliyotda keng qo'llanilishiga olib keladi.

Foydalanilgan adabiyotlar

1. Karp, R. M. (1991). An introduction to randomized algorithms. Discrete 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. С#. 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. [Raxmonjonovich, F. S. \(2024\).](#) MA'LUMOTLARNI SIQISHDA BITLI ALGORITMLARDAN FOYDALANISH. Modern education and development, 15(5), 320-328.
10. [Raxmonjonovich, F. S. \(2024\).](#) AXBOROTLARNI SHIFRLASHDA МАТЕМАТИК ALGORITMLARDAN FOYDALANISH. Modern education and development, 15(5), 338-344.
11. [Raxmonjonovich, F. S. \(2024\).](#) BIR SHAHARDAN BOSHQASIGA YUK YETKAZIB BERISHDA ENG OPTIMAL VA KAM XARAJAT SARFLANADIGAN YO'LNI TOPISHDA BELLMAN-FORD ALGORITMIDAN FOYDALANISH. Ta'lim innovatsiyasi va integratsiyasi, 34(2), 72-78.
12. [Raxmonjonovich, F. S. \(2024\).](#) KOMPYUTER TARMOQLARI SOHASIDA BITLI ALGORITMLAR. Modern education and development, 15(4), 50-59.
13. [Raxmonjonovich, F. S., & Xurshidbek o'g'li, A. O. \(2024\).](#) FORD-BELMAN ALGORITMI. Modern education and development, 15(4), 60-65.
14. [Raxmonjonovich, F. S. \(2024\).](#) IJTIMOYIY TARMOQLAR TAHLILIDA BFS ALGORITMLARI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 58(7), 20-26.
15. [Raxmonjonovich, F. S. \(2024\).](#) DINAMIK DASTURLASH VA TARMOQ OQIMIDA FORD-BELMAN ALGORITMIDAN FOYDALANISH. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 58(7), 13-19.
16. [Raxmonjonovich, F. S. \(2024\).](#) GRAFLARDA FLOYD-WARSHALL ALGORITMINING AHAMIYATI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 58(7), 6-12.
18. [Raxmonjonovich, F. S., & Hamdamjon o'g'li, A. S. \(2024\).](#) HISOBLASH МАТЕМАТИКАСИ VA SONLI ANALIZ SOHASIDA DIFFERENSIAL TENGLAMALARNI YECHISHDA МАТЕМАТИК ALGORITMLARNING AHAMIYATI. TADQIQOTLAR. UZ, 51(2), 37-44.
19. [Raxmonjonovich, F. S. \(2024\).](#) ARIFMETIK VA GEOMETRIK PROGRESSIYAGA OID MASALALARNING МАТЕМАТИК ALGORITMLARI YORDAMIDA YECHISH. Лучшие интеллектуальные исследования, 34(1), 142-152.
20. [Raxmonjonovich, F. S. \(2024\).](#) XOFMAN KODLASH TIZIMI: AVIATSIYA VA PARVOZ MA'LUMOTLARINI SIQISHNING INNOVATSION YONDASHUVI. Лучшие интеллектуальные исследования, 34(1), 153-160.
21. [Raxmonjonovich, F. S., & Botirali o'g'li, T. M. \(2024\).](#) KAN ALGORITMINI GRAFLARDA QO'LLANILISHI. TADQIQOTLAR. UZ, 51(2), 27-36.

22. Raxmonjonovich, F. S. (2024). ROBOTOTEXNIKA SOHASIDA GEOMETRIK ALGORITMLARNING O'RNI. Лучшие интеллектуальные исследования, 34(1), 134-141.

23. Raxmonjonovich, F. S. (2024). MINIMAL BOG'LANISH DARAXTINI TOPISHDA PRIM ALGORITMIDAN FOYDALANISH. YANGI O'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(3), 436-443.

24. Raxmonjonovich, F. S., & Kudratullo o'g, K. U. B. (2024). C# VA NET FRAMEWORK ORQALI ZAMONAVIY VA XAVFSIZ TARMOQ DASTURLARINI ISHLAB CHIQUISH. International journal of scientific researchers (IJSR) INDEXING, 5(2), 351-356.

25. Raxmonjonovich, F. S., & Azizjon o'g'li, N. A. (2024). WORKING WITH DATE AND TIME IN MODERN PROGRAMMING LANGUAGES. International journal of scientific researchers (IJSR) INDEXING, 5(2), 296-300.

26. Farmonov, S., & Rustamova, N. (2024, May). SINFLASHNING METRIK ALGORITMLARI, YAQIN QO'SHNI USULI VA UNI UMUMLASHTIRISH HAMDA ULARNI NEYRON TARMOQ TEXNOLOGIYALARIDA QO'LLANILISHI. In Международная конференция академических наук (Vol. 3, No. 5, pp. 71-75).

27. Farmonov, S., & Ergashaliyeva, B. (2024). QATIYMAS NEYRON TO'RLAR: MAMDANI QATIYMAS MANTIQIY XULOSASI, SUGENO QATIYMAS MANTIQIY XULOSASI. Development and innovations in science, 3(5), 62-70.

28. Raxmonjonovich, F. S. (2024). KOMPYUTER GRAFIKASI VA O'YIN DASTURLASHDA JOHNSON ALGORITMINING AHAMIYATI. Ta'lim innovatsiyasi va integratsiyasi, 34(2), 145-151.

29. Farmonov, S. R., & qizi Oktamjonova, M. I. (2024, November). FLOYD-UORSHELL ALGORITMI. In International Conference on World Science and Resarch (Vol. 1, No. 3, pp. 32-42).

30. Raxmonjonovich, F. S. (2024). KAN ALGORITMI VA UNING AMALIY QO'LLANILISHLARI. Ta'lim innovatsiyasi va integratsiyasi, 34(2), 139-144.

31. Farmonov, S. R. (2024). BFS ALGORITMI ORQALI TOPOLOGIK TARTIBNI ANIQLASH. ОБРАЗОВАНИЕ И НАУКА В XXI BEKE, (56-5).

31. Raxmonjonovich, F. S., & Saidahmad o'g'li, I. S. (2024). BFS ALGORITMI VA UNING XAVFSIZLIK SOHASIDAGI ROLI. SCIENTIFIC APPROACH TO THE MODERN EDUCATION SYSTEM, 3(31), 117-123.