SUN'IY INTELLEKT VA MASHINALARGA OID MASALALARNI BFS ALGORITIMI YORDAMIDA YECHISH

Farmonov Sherzodbek Raxmonjonovich
Farg'ona davlat universiteti amaliy matematika va Informatika kafedrasi katta o'qituvchisi
famonovsh@gmail.com
Aminjonova Dilobarxon Baxtiyor qizi
Farg'ona davlat univarsiteti talabasi
aminjonivadilobar@gmail.com

Annotatsiya. Ushbu maqolada Bfs algoritimidan foydalanish mumkin bo'lgan sohalar, shu sohalardagi qo'llash mumkin bo'lgan o'rinlar keltirilgan bo'lib sun'iy intellekt va mashinalar bilan ishlashga doir misol va masalalarni Bfs algoritimi yordamida samarali yechish va tahlil qilish ko'rib chiqilgan.

Kalit so'zlar: Bfs algoritimi, sun'iy intellekt, grafda harakat,kenglik bo'yicha qidiruv, o'yinlarda yo'l topish, eng qisqa yo'l topish.

Annotation. This article lists the areas in which the BFS algorithm can be used, the areas that can be applied in these areas, and the effective solution and analysis of examples and issues of working with artificial intelligence and machines using the BFS algorithm.

Key words: Bfs algorithms, artificial intelligence, action in the graph, search by latitude, find your way in games, find the shortest path.

Аннотация. В данной статье представлены области, в которых может использоваться алгоритм Bfs, возможные места применения в этих областях, а также рассмотрены примеры и проблемы, связанные с работой с искусственным интеллектом и машинами, а также эффективное решение и анализ проблем с использованием рассмотрен алгоритм Bfs.

Ключевые слова: алгоритм Bfs, искусственный интеллект, обход графа, поиск по широте, поиск пути в играх, поиск кратчайшего пути.

BFS (Breadth-First Search) algoritmi - graf yoki daraxt (tree) bo'ylab kenglik bo'yicha qidirish algoritmi. Ushbu algoritm grafning barcha tugunlarini (yoki faqat kerakli tugunni) bosqichma-bosqich qidiradi, har bir darajadagi tugunlarni navbatma-navbat tekshiradi.Bfs algoritimining ishlash jarayonini quyidagicha:**Boshlanish nuqtasi:** Algoritm birinchi tugundan (odatda boshlang'ich tugun) boshlanadi va uni tashrif buyurilgan deb belgilaydi.**Navbatdan foydalanish:** BFS algoritmi tugunlarni saqlash va tekshirish uchun **FIFO (First In, First Out)** prinsipiga asoslangan navbatdan foydalanadi.**Qo'shnilarni tekshirish:** Har bir tugun uchun uning barcha

qo'shnilari (bog'langan tugunlar) navbatga qo'shiladi, agar ularga hali tashrif buyurilmagan bo'lsa.**Tugatish:** Algoritm barcha tugunlarni qidirgach yoki kerakli tugunni topgach, yakunlanadi.

BFS (Breadth-First Search) algoritmi bir qancha sohalarda qo'llaniladi, jumladan:

1. Grafik va tarmoqlarni qidirish: BFS algoritmi grafda qisqa yo'llarni topish uchun ishlatiladi. Masalan, ijtimoiy tarmoqlarda foydalanuvchilar orasidagi eng qisqa bog'lanishni aniqlash uchun ishlatilishi mumkin.

2. Yo'l topish va xaritalash: BFS turli yo'l topish tizimlarida, masalan, GPS va navigatsiya dasturlarida eng qisqa yo'lni topishda ishlatiladi.

3. O'yinlar va muammolarni yechish: BFS algoritmi labirint kabi o'yinlarda yoki jumboqlarda chiqish yo'lini topish uchun ishlatiladi.

4. Tarmoqlarda zararlanishlarni aniqlash: BFS yordamida tarmoqlarda infektsiyani aniqlash va uni izolyatsiya qilishda foydalaniladi.

5. Sun’iy intellekt va mashina o’rganish: BFS ba'zi AI tizimlarida optimal strategiyalarni izlashda va muammolarni yechishning optimal usullarini topishda qo'llaniladi.

6. Ma'lumotlarni indekslash va qidirish: Ma'lumotlarni indekslashda va katta hajmdagi ma'lumotlar ichidan izlashda qo'llaniladi, masalan, fayl tizimlarida fayllarni qidirishda.

Bu sohalarda BFS algoritmi samaradorligi va oddiyligi tufayli keng qo'llaniladi.

Sun'iy intellekt va mashina o'rganishda BFS algoritmi bir qator muhim masalalarni hal qilishda ishlatiladi.Quyida ularning asosiy qo’llanilishi keltirilgan:

1. Eng qisqa yo'lni topish: Grafga asoslangan AI masalalarida, masalan, o'yinlarda harakat yo'nalishini aniqlashda yoki robotikada eng qisqa yo'lni topishda BFS algoritmidan foydalaniladi.

2. Holatlarni tadqiq qilish: Jumboqlarni yechish (masalan, “15 ta kvadrat” kabi jumboqlar) yoki holatlarni izlash (o'yinlarda harakatni rejalashtirish) kabi masalalarda barcha mumkin bo'lgan holatlarni tekshirish uchun BFS yordam beradi.

3. O'yinlarda harakat va strategiya izlash: BFS orqali AI agenti o'yinda optimal harakatlarni aniqlashi mumkin, masalan, o'yinchining maqsadga yetib borishi uchun eng samarali yo'lni topish.

4. Grafga asoslangan muammolarni yechish: Grafik tahlil qilish talab qilinadigan masalalarda (masalan, ijtimoiy tarmoqlardagi foydalanuvchi aloqalarini o'rganish) BFS yordamida foydalanuvchilar yoki ob'ektlar orasidagi bog'lanishlarni va o'zaro ta'sirlarni tahlil qilish mumkin.

5. Tarmoqlarda virus tarqalishini aniqlash: Virus yoki zararli kod tarqalishini simulyatsiya qilishda va uni to'xtatishda BFS bilan virusning barcha ehtimoliy tarqalish yo'nalishlarini tekshirish mumkin.

6. Tabiiy tilni qayta ishlashda (NLP): Ba'zi holatlarda, masalan, gap tahlili yoki

matn tarkibini aniqlashda, grafga asoslangan tuzilmalar bilan ishlaganda BFS yordamida optimal yo'llarni topish yoki turli bog'lanishlarni tahlil qilish mumkin.

Ushbu masalalarda BFS algoritmi tezkor va oddiy ishlashi sababli foydalaniladi. Quyida BFS algoritmiga oid masala keltirilgan:

Masala: Labirintdan chiqish yo'lini topish

5x5 o‘lchamdagi labirint mavjud, unda:

"1" – ochiq yo‘l (harakat qilish mumkin bo‘lgan hujayra),

"0"– to‘silgan joy (harakat qilib bo‘lmaydigan hujayra) deb qabul qilinadi.

Labirintning boshlanish nuqtasi yuqori chap burchakda joylashgan, ya'ni koordinatasi (0, 0) va chiqish nuqtasi pastki o‘ng burchakda joylashgan, ya'ni koordinatasi (4, 4).Labirintning boshlanish nuqtasidan chiqish nuqtasigacha eng qisqa yo‘lni toping. Harakat qilish mumkin bo‘lgan yo‘nalishlar:

1. yuqoriga,
2. pastga,
3. chapga,
4. o‘ngga.

Labirint quyidagicha:

1 1 0 1 1
0 1 0 1 0
0 1 1 1 0
0 0 0 1 0
1 1 1 1 1

Yechim:

BFS algoritmi yordamida labirintda har bir nuqtani ketma-ket tekshirib, eng qisqa yo'lni topishimiz mumkin.

1. Boshlang'ich holatni tanlash: (0, 0) nuqtasidan boshlaymiz va u yerdan barcha qo'shni nuqtalarni tekshiramiz.

2. Navbatga qo'shish: Har bir nuqtani navbatga qo'shamiz va ular orqali harakat qilamiz. Agar (4, 4) nuqtasiga yetib borsak, yo'l topilgan bo'ladi.

3. Tekshirish yo'nalishlari: Biz har bir nuqtadan to'rtta yo'nalishda harakat qilishimiz mumkin: yuqoriga, pastga, chapga va o'ngga.

Yechimni toppish:

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

class Program
{
   static int[,] maze = {
```

```
        { 1, 1, 0, 1, 1 },
        { 0, 1, 0, 1, 0 },
        { 0, 1, 1, 1, 0 },
        { 0, 0, 0, 1, 0 },
        { 1, 1, 1, 1, 1 }
    };

    static int n = 5, m = 5
    static int[] dx = { -1, 1, 0, 0 };
    static int[] dy = { 0, 0, -1, 1 };


    static int BFS(Tuple<int, int> start, Tuple<int, int> end)
    {
        Queue<Tuple<int, int, int>> queue = new Queue<Tuple<int, int, int>>();
        bool[,] visited = new bool[n, m];

        queue.Enqueue(new Tuple<int, int, int>(start.Item1, start.Item2, 0));
        visited[start.Item1, start.Item2] = true;

        while (queue.Count > 0)
        {
            var node = queue.Dequeue();
            int x = node.Item1;
            int y = node.Item2;
            int dist = node.Item3;

            if (x == end.Item1 && y == end.Item2)
                return dist;

            for (int i = 0; i < 4; i++)
            {
                int nx = x + dx[i];
                int ny = y + dy[i];
                if (nx >= 0 && nx < n && ny >= 0 && ny < m && maze[nx, ny] == 1
&& !visited[nx, ny])
                {
                    visited[nx, ny] = true;
                    queue.Enqueue(new Tuple<int, int, int>(nx, ny, dist + 1));
```

```
                }
            }
        }

        return -1;
    }

    static void Main()
    {
        var start = new Tuple<int, int>(0, 0);
        var end = new Tuple<int, int>(4, 4);

        int distance = BFS(start, end);

        if (distance != -1)
            Console.WriteLine("Eng qisqa yo'l: " + distance);
        else
            Console.WriteLine("Yo'l topilmadi");
    }
}
```

Natija:

Ushbu kod “Eng qisqa yo'l: 8” deb chiqaradi, ya'ni labirintda boshlanish nuqtasidan chiqish nuqtasigacha bo'lgan eng qisqa yo'l 8 qadamdan iborat.

Tushuntirish:

- Har bir nuqtani tekshirayotganda, navbatga qo'shib boramiz va o'sha nuqtadan eng qisqa masofani hisoblaymiz.

- Agar (4, 4) nuqtasiga yetib borsak, masofani qaytaramiz. Agar yo'l bo'lmasa, -1 qiymatini qaytaradi.

Bu masala BFS algoritmi yordamida qanday qilib eng qisqa yo'lni topishni amalda ko'rsatadi.

BFS algoritmi dasturlashda keng qo'llaniladigan va ko'p muammolarni yechishda asosiy vosita hisoblanadi. U grafdagi eng qisqa yo'lni topish, ijtimoiy tarmoq tahlillari va murakkab tizimlarda optimal yo'nalishlarni aniqlashda samarali yechim beradi. FIFO tamoyiliga asoslanganligi uning ishlashini oson tushuniladigan va sodda qiladi. Shuningdek, BFS algoritmi dasturchilar uchun graf va daraxt tuzilmalarini o'rganishning muhim asosi bo'lib xizmat qiladi. Uni o'zlashtirish orqali murakkab muammolarni yechish qobiliyatini rivojlantirish mumkin. Shunday qilib, Bfs algoritimlari sun’iy intellekt va mashinalar bilan ishlashda foydali va samarali bo’lishi mumkin.

Foydalanilgan adabiyotlar ro'yxati:

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. B CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (Т. 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'EKTGA 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.