

## KAN ALGORITMINI GRAFLARDA QO'LLANILISHI

**Farmonov Sherzodbek Raxmonjonovich**

Farg'ona davlat universiteti amaliy  
matematika va informatika kafedrasida katta o'qituvchisi  
[farmonovsh@gmail.com](mailto:farmonovsh@gmail.com)

**Tojialiyev Muhammadali Botirali o'g'li**

Farg'ona davlat universiteti 2-kurs talabasi  
[tojialiyevmuhammadali@gmail.com](mailto:tojialiyevmuhammadali@gmail.com)

**Annotatsiya:** Ushbu maqola yo'naltirilgan grafda eng qisqa yo'lni topish uchun Kan algoritmini qanday qo'llashni ko'rsatadi. Kan algoritmi, odatda, topologik saralashda ishlatiladi, lekin bu maqolada uni eng qisqa yo'lni hisoblash maqsadida qo'llash yondoshuvi ta'riflanadi. Maqolada C# dasturlash tilida yozilgan kod orqali ushbu algoritmnining ishlash printsipti tushuntirilgan va misollar bilan ko'rsatilgan. Ushbu yondoshuvni **Kan algoritmi** yordamida eng qisqa yo'lni topish uchun ishlatish, masalaning yechimini yanada samarali va tushunarli qiladi.

**Kalit so'zlar:** Kan algoritmi, topologik saralash, yo'naltirilgan graf, eng qisqa yo'l, siklsiz graf, C# dasturlash, topologik tartib, masofa hisoblash.

**Abstract:** This paper shows how to use Kahn's algorithm to find the shortest path in a directed graph. Kahn's algorithm is commonly used in topological sorting, but this article describes an approach to apply it to shortest path computation. The article explains the working principle of this algorithm through the code written in the C# programming language and shows examples. Using this approach to find the shortest path using Kahn's algorithm makes the solution of the problem more efficient and understandable.

**Key words:** Kahn's algorithm, topological sorting, directed graph, shortest path, acyclic graph, C# programming, topological ordering, distance calculation.

**Аннотация:** В этой статье показано, как использовать алгоритм Кана для поиска кратчайшего пути в ориентированном графе. Алгоритм Кана обычно используется при топологической сортировке, но в этой статье описывается подход к его применению для вычисления кратчайшего пути. В статье объясняется принцип работы этого алгоритма на примере кода, написанного на языке программирования C#, и приводятся примеры. Использование такого подхода для поиска кратчайшего пути с помощью алгоритма Кана делает решение задачи более эффективным и понятным.

**Ключевые слова:** алгоритм Кана, топологическая сортировка, ориентированный граф, кратчайший путь, ациклический граф,

программирование на C#, топологическое упорядочение, вычисление расстояния.

**Kirish: Kan Algoritmi va hal qilinishi kerak bo'lgan muammo haqida:** Kan algoritmi, yo'nalgan grafda tugunlarni topologik saralash uchun ishlatiladi. Agar grafda  $u \rightarrow v$  kabi yo'nalish mavjud bo'lsa, unda  $u$  tuguni  $v$  tugunidan oldin joylashishi kerak. Kan algoritmi bu tartibni qanday qilib samarali ravishda topishni ko'rsatadi. Yo'nalgan grafda barcha tugunlar bir-biriga bog'langan bo'lishi mumkin, ba'zida esa tugunlar orasida bog'lanishlar bo'lmasligi ham mumkin. Topologik saralashda eng asosiy shartlardan biri bu grafda siklning yo'qligidir. Agar grafda tsikl bo'lsa, topologik saralashni amalga oshirish mumkin bo'lmaydi, chunki bunday holda tugunlarning tartibini aniq belgilab bo'lmaydi.

**Kan algoritmi quyidagi sohalarda keng qo'llaniladi:**

#### 1. Raqamli Elektronika va Logika Dizayni

Kan algoritmi eng ko'p raqamli elektronika va logika dizayn sohasida ishlatiladi. Bu sohada, masalan:

- Mantiqiy sxemalar yaratish: Raqamli tizimlarda mantiqiy funksiyalarni soddalashtirish orqali kompyuterlar, mikrosxemalar va boshqa raqamli qurilmalar uchun samarali logik sxemalar quriladi. Bu, o'z navbatida, tizimlarning tezligini oshiradi va energiya sarfini kamaytiradi.
- Integratsiyalashgan sxemalar (ICs): Kan xaritalari mikrochiplar va integratsiyalashgan sxemalar (IC) dizaynida mantiqiy funksiyalarni soddalashtirishda ishlatiladi. Masalan, ko'plab logik eshiklar (AND, OR, NOT va boshqalar) integratsiyalashgan chiplar yordamida yig'iladi.

#### 2. Kompyuter Arxitekturasi va Dasturlash

- Raqamli hisoblash tizimlari: Kan algoritmi, masalan, ko'plab kirishlarga ega bo'lgan mantiqiy funksiyalarni minimal ifodalarda yozishda foydalidir, bu esa yuqori samarali hisoblash tizimlarini yaratishga yordam beradi.
- Tizim optimizatsiyasi: Tizimlar samaradorligini oshirish maqsadida mantiqiy funksiyalarni soddalashtirish orqali kompyuter arxitekturasi optimallashtiriladi.
- Tizim dizaynida: C# yoki boshqa dasturlash tillarida mantiqiy funksiyalarni dasturlashda Kan algoritmiga tayanish mumkin. Bunday holda, dastur optimizatsiyasi va resurslardan samarali foydalanish uchun funksiyalarni soddalashtirish kerak bo'ladi.

#### 3. Kompilyatorlar va Matematika

- Kompilyatorlar dizayni: Kompilyatorlarda mantiqiy ifodalarni optimallashtirishda Kan xaritalari qo'llaniladi. Bu, masalan, ma'lumotlar oqimining samaradorligini oshirishga yordam beradi va ma'lumotlarni tezroq ishlov berishga imkon yaratadi.

- Logik ifodalarning soddalashtirilishi: Kan algoritmi, matematikada yoki boshqa ilmiy sohalarda, mantiqiy formulalar va ifodalarni qisqartirishda ishlatiladi, bu esa hisob-kitoblarni tezlashtiradi.

#### 4. Avtomatlashtirish va Robototexnika

- Avtomatlashtirilgan tizimlarda qaror qabul qilish: Avtomatik tizimlarda qaror qabul qilish jarayonida Kan algoritmi yordamida tizimdagi mantiqiy funksiyalar soddalashtiriladi. Bu esa tizimning ishlash tezligini oshiradi va resurslardan yanada samarali foydalanishni ta'minlaydi.
- Robotlarning boshqaruv tizimlari: Robototexnikada ham Kan xaritalaridan foydalanish mumkin. Robotlar uchun mantiqiy boshqaruv sxemalarini yaratishda va ularni optimallashtirishda Kan algoritmi yordamida kompleks boshqaruv tizimlarini yaratish mumkin.

#### 5. Sun'iy Intellekt (AI) va Mashinani O'rganish

- Optimizatsiya va qaror qabul qilish: Kan algoritmi, sun'iy intellekt tizimlarida qaror qabul qilish jarayonlarini optimallashtirish uchun ishlatilishi mumkin. Mantiqiy funksiyalarni soddalashtirish orqali tizimning samaradorligini oshirish mumkin.
- Mashina o'rganishda: Ayniqsa, mashina o'rganish modellarini optimallashtirishda mantiqiy funksiyalarni soddalashtirish va resurslarni samarali taqsimlashda Kan algoritmiga asoslanish mumkin.

#### 6. Testlash va Xatoliklarni Izlash (Debugging)

- Test sxemalari yaratish: Dasturlashda, tizimlar va sxemalar test qilish jarayonida, mantiqiy funksiyalarni optimallashtirish, test uchun kerakli shartlarni topishda Kan algoritmi yordam beradi.
- Xatoliklarni aniqlash: Kod yoki sxemada xatoliklarni aniqlashda Kan xaritalari yordamida mantiqiy jarayonlar va xatolarni vizual tarzda tahlil qilish mumkin.

#### 7. Telekommunikatsiya va Signalni Ishlash

- Signalni kodlash va dekodlash: Kan algoritmi telekommunikatsiya tizimlarida, masalan, signalni kodlash va dekodlash jarayonlarini optimallashtirishda ishlatiladi.
- Mantiqiy analizlar: Telekommunikatsiya tizimlarida ma'lumotlar uzatish va qayta ishlashda mantiqiy analiz va optimizatsiya kerak bo'lganda Kan algoritmi yordam berishi mumkin.

#### 8. Yapay Zeka va Fikrni Shakllantirish

Kan algoritmi yordamida dasturlashda yoki raqamli tizimlar yaratishda, sun'iy intellekt (AI) tizimlarining ishlashini optimallashtirish va mantiqiy qarorlarni soddalashtirishga yordam beradi.

**Kan algoritmini qo'llash mumkin bo'lgan sohalardan biri bu- Graflardir**

Graf nazariyasida topologik saralash, yo‘nalishli grafdagi tugunlarni tartibga solishning muhim usulidir. Bu jarayon, tugunlar orasidagi yo‘nalgan qirralarning mavjudligini hisobga olib, tugunlarni aniq tartibda joylashtirishni anglatadi. Topologik saralashning amaliy qo‘llanilishi juda keng bo‘lib, unda eng ko‘p uchraydigan misollar — vazifalarni bajarish tartibini belgilash, topshiriqlarni prioritetga ko‘ra tartibga solish va resurslarni boshqarishdir.

Topologik saralashni amalga oshirishning eng samarali usullaridan biri Kan algoritmi (yoki Kan's Algorithm)dir. Bu maqolada, Kan algoritmining mohiyati, uning ishlash prinsipi va qo‘llanilish sohalari haqida batafsil tushuntiriladi.

**Yandex** ilovalari, ayniqsa **Yandex Maps** va **Yandex Navigator** kabi xizmatlar, graflardan foydalanadi. Bu turdagi xizmatlar, asosan, **yo'naltirilgan graflar** (directed graphs) prinsiplariga asoslangan bo‘lib, ular orqali yo‘nalishlarni topish, manzillarni xaritada ko‘rsatish va transport tizimlarini boshqarish mumkin.

### **Yandex ilovalari qanday graflardan foydalanadi?**

#### **1. Yo'naltirilgan Transport Graflari (Directed Graphs):**

- Yandex Maps va Yandex Navigator kabi ilovalar **yo'naltirilgan graflarni** qurishadi. Bu graflarda tugunlar (nodes) — **ko'chalardagi chorrahalar, yo'llar** yoki **manzillar** bo‘lib, qirralar (edges) esa **yo'nalishlar** yoki **yo'llardagi yo'nalishlarni** ifodalaydi.
- Har bir yo‘nalish (masalan, ko‘cha yoki yo‘l) yo‘naltirilgan bo‘lishi mumkin. Bu, masalan, bir yo‘lda faqat bir yo‘nalishda harakatlanish mumkinligini anglatadi. Shunday qilib, har bir yo‘l va ko‘cha o‘z yo‘nalishiga ega.

#### **2. Yo'nalishlar va Traffic (Trafik) Analiz:**

- Yandex Navigator ilovasi uchun trafik tahlilini amalga oshirishda, foydalanuvchining manziliga borish uchun eng yaxshi yo‘lni topish uchun **yo'naltirilgan graflardan** foydalaniladi.
- Yandex algoritmlari har bir yo‘l va chorraha o‘rtasidagi trafikni (masalan, og‘irlik darajasi yoki tiqilinch holati) hisobga olib, eng optimal yo‘nalishni taklif qiladi.

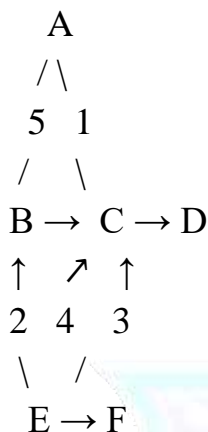
#### **3. Dastlabki Yo'nalishlarni Qidirish:**

- Masalan, **Google Maps** yoki **Yandex Maps** kabi xarita ilovalari foydalanuvchiga manzilga yetib borish uchun eng qisqa yoki eng tezkor yo‘lni topishda **graf algoritmlarini** qo‘llaydi. Eng mashhur algoritmlar orasida **Dijkstra algoritmi** yoki **A (A-star)\*** algoritmi mavjud. Bu algoritmlar graflardagi tugunlar o‘rtasidagi eng qisqa yo‘lni topish uchun ishlatiladi.

#### **4. Qattiq va O'zgaruvchan Trafik Sharoitlari:**

- Yandex ilovalari, shuningdek, real vaqt rejimida mavjud trafik holatini tahlil qilib, yo‘nalishni va graflarni **dinamik ravishda** yangilaydi. Masalan, agar biror yo‘lda tiqilinch bo‘lsa, grafdagi qirra vazifasini bajaradigan yo‘lni **avtomatik tarzda** yangilaydi.

Ushbu grafni **Kan algoritmi** yordamida topologik saralash:



- $A \rightarrow B$  (5 km)
- $A \rightarrow C$  (1 km)
- $B \rightarrow E$  (2 km)
- $C \rightarrow D$  (4 km)
- $C \rightarrow F$  (3 km)
- $D \rightarrow F$  (3 km)
- $E \rightarrow F$  (1 km)

**Kan algoritmi yordamida topologik saralash:**

**1. Boshlang'ich grafiga qo'shni tugunlar va kirish darajalarini hisoblash:**

- $A$  tugunining kirish darajasi 0, chunki unga hech narsa yo'naltirilmagan.
- $B$  tugunining kirish darajasi 1, chunki  $A \rightarrow B$  qirradi mavjud.
- $C$  tugunining kirish darajasi 1, chunki  $A \rightarrow C$  qirradi mavjud.
- $D$  tugunining kirish darajasi 1, chunki  $C \rightarrow D$  qirradi mavjud.
- $E$  tugunining kirish darajasi 1, chunki  $B \rightarrow E$  qirradi mavjud.
- $F$  tugunining kirish darajasi 3, chunki  $C \rightarrow F$ ,  $D \rightarrow F$ , va  $E \rightarrow F$  qirralari mavjud.

**2. Topologik saralash jarayoni:**

1. Boshlang'ichda kirish darajasi 0 bo'lgan tugunlarni navbatga qo'yamiz.
2.  $A$  tuguni kirish darajasi 0 bo'lgan tugun, shuning uchun uni navbatga qo'yamiz.

Navbat: [A]

**3. A tugunidan chiqadigan yo'nalishlar:**

1.  $A \rightarrow B$  va  $A \rightarrow C$  yo'nalishlari mavjud. Bu yo'nalishlar orqali  $B$  va  $C$  tugunlarining kirish darajalari kamayadi.
2.  $B$  tugunining kirish darajasi 0 bo'lsa, uni navbatga qo'yamiz.
3.  $C$  tugunining kirish darajasi 0 bo'lsa, uni navbatga qo'yamiz.

Navbat: [A, B, C]

**4. B tugunidan chiqadigan yo'nalish:**

1.  $B \rightarrow E$  yo'nalishi mavjud, shu orqali  $E$  tugunining kirish darajasi kamayadi. Endi  $E$  tugunining kirish darajasi 0 bo'ladi, shuning uchun uni navbatga qo'yamiz.

Navbat: [A, B, C, E]

#### 5. C tugunidan chiqadigan yo'nalishlar:

1.  $C \rightarrow D$  va  $C \rightarrow F$  yo'nalishlari mavjud. Bu yo'nalishlar orqali  $D$  va  $F$  tugunlarining kirish darajalari kamayadi.
2.  $D$  tugunining kirish darajasi 0 bo'lsa, uni navbatga qo'yamiz.
3.  $F$  tugunining kirish darajasi 0 bo'lsa, uni navbatga qo'yamiz.

Navbat: [A, B, C, E, D, F]

#### 6. D tugunidan chiqadigan yo'nalish:

1.  $D \rightarrow F$  yo'nalishi mavjud, ammo  $F$  tuguni allaqachon navbatda bor, shuning uchun bu yo'nalishdan keyin hech narsa o'zgaradi.

Topologik saralash natijasi: [A, B, C, E, D, F]

Shu misolni C# dagi kodi:

```
using System;
```

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

```
class KahnAlgorithm
```

```
{
```

```
    static Dictionary<int, List<int>> graph = new Dictionary<int, List<int>>();
```

```
    static Dictionary<int, int> inDegree = new Dictionary<int, int>();
```

```
    public static void AddEdge(int from, int to)
```

```
{
```

```
        if (!graph.ContainsKey(from))
```

```
            graph[from] = new List<int>();
```

```
        graph[from].Add(to);
```

```
        if (!inDegree.ContainsKey(to))
```

```
            inDegree[to] = 0;
```

```
        inDegree[to]++;
```

```
        if (!inDegree.ContainsKey(from))
```

```
            inDegree[from] = 0;
```

```
}
```

```
public static List<int> TopologicalSort()
```

```
{
```

```
    Queue<int> queue = new Queue<int>();
```

```
    List<int> topologicalOrder = new List<int>();
```

```

foreach (var node in inDegree)
{
    if (node.Value == 0)
        queue.Enqueue(node.Key);
}
while (queue.Count > 0)
{
    int currentNode = queue.Dequeue();
    topologicalOrder.Add(currentNode);

    if (graph.ContainsKey(currentNode))
    {
        foreach (var neighbor in graph[currentNode])
        {
            inDegree[neighbor]--;
            if (inDegree[neighbor] == 0)
                queue.Enqueue(neighbor);
        }
    }
}

return topologicalOrder;
}
static void Main()
{
    // Grafiga qirralarni qo'shish (A=1, B=2, C=3, D=4, E=5, F=6)
    AddEdge(1, 2); // A → B
    AddEdge(1, 3); // A → C
    AddEdge(2, 5); // B → E
    AddEdge(3, 4); // C → D
    AddEdge(3, 6); // C → F
    AddEdge(4, 6); // D → F
    AddEdge(5, 6); // E → F
    var result = TopologicalSort();
    Console.WriteLine("Topologik saralash:");
    foreach (var node in result)
    {
        Console.Write(node + " ");
    }
}

```

}  
 }  
 }

### Kan algoritmining vaqt murakkabligi $O(V + E)$ bo'lib, bunda:

- ❖  $V$  — grafdagi tugunlar soni.
- ❖  $E$  — grafdagi qirralar soni.

Kan algoritmi, topologik saralashni amalga oshirishda samarali va ishonchli usuldir. U turli xil vazifalarni bajarish tartibini belgilashda, resurslarni boshqarishda va dasturiy ta'minotlarda qo'llaniladi. Kan algoritmining afzalliklari uning yuqori samaradorligi, soddaligi va  $O(V + E)$  vaqt murakkabligi hisoblanadi. Agar graf siklsiz bo'lsa, Kan algoritmi eng yaxshi tanlovdir.

### 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.Эрикссон. 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.
7. Рафгарден Тим. Совершенный алгоритм. Графовые алгоритмы и структуры данных. – СПб.: Питер, 2019. - 256 с.
8. [Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э.](#) Структуры данных и алгоритмы. – М.: [Вильямс](#), 2018. – 400 с.
9. 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.
10. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. В CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (Т. 2, Выпуск 12, сс. 71–74). Zenodo.
11. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. *Theoretical aspects in the formation of pedagogical sciences*, 2(22), 90-96
12. 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.



13. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 425-429.
14. Raxmonjonovich, F. S. (2023). Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 434-438.
15. 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.
16. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 475-481.
17. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 482-486.
18. 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.
19. 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.
20. Raxmonjonovich, F. S. (2023). C# tilida ArrayList bilan ishlashning afzalliklari. *Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari*, 2(2), 470-474.
21. 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>.
22. 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.
23. Farmonov, S., & Rasuljonova, Z. (2024). OB'EKTGA YO'NALTIRILGAN DASTURLASH ZAMONAVIY DASTURLASHNING ASOSI SIFATIDA. *Центральноазиатский журнал образования и инноваций*, 3(1), 83-86.
24. 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.

25. 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.
26. Raxmonjonovich, F. S. (2024). C# VA MASHINA TILI. *Ta'lim innovatsiyasi va integratsiyasi*, 12(1), 59-62.
27. Farmonov, S. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. *Центральноазиатский журнал образования и инноваций*, 2(12 Part 2), 71-74.
28. Farmonov, S., & Jo'rayeva, M. (2023, December). DASTURLASHDA POLIMORFIZMNING AHAMIYATI. In *Международная конференция академических наук* (Vol. 2, No. 13, pp. 5-8).
29. Farmonov, S., & Usmonaliyev, U. (2024). O'ZBEKISTON RESPUBLIKASI IT SOHASINING RIVOJLANISH ISTIQBOLLARI. *Бюллетень педагогов нового Узбекистана*, 2(1), 59-62.
30. 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.