

## KAN ALGORITMI ASOSIDA GRAFLARDA SIKLSIZ TARTIBNI ANIQLASH

***Farmonov Sherzod Raxmonjonovich***

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

***Yoqubova Shahnoza Rustamjon qizi***

*Farg'ona davlat universiteti Amaliy matematika yo'nalishi 2-kurs  
talabasi [syoqubova06@gmail.com](mailto:syoqubova06@gmail.com)*

***Annotatsiya.*** Ushbu maqola, yo'nalgan atsiklik graflarda (DAG) topologik tartibni aniqlash uchun keng qo'llaniladigan Kahn algoritmini o'rganadi. Kahn algoritmi, graflarning siklsiz tuzilishlarini tekshirish va ularning tugunlarini samarali tarzda tartiblashni ta'minlash uchun ishlatiladi. Maqolada, algoritmning ishlash printsiplari, uning asosiy qadamlarini batafsil tahlil qilish, shuningdek, algoritmning samaradorligi va vaqt murakkabligi masalalari ko'rib chiqiladi. Kahn algoritmi, har bir tugunning kirish darajasini (in-degree) hisoblash orqali graflarda sikl mavjudligini aniqlashga yordam beradi.

***Kalit so'zlar:*** Kahn algoritmi, topologik tartib, yo'nalgan atsiklik graf (DAG), sikl aniqlash, graf tahlili, algoritm samaradorligi.

***Аннотация:*** В этой статье исследуется алгоритм Кана, который широко используется для определения топологического порядка в ориентированных ациклических графах (DAG). Алгоритм Кана используется для проверки ациклической структуры графов и обеспечения эффективного упорядочения их узлов. В статье рассмотрен принцип работы алгоритма, подробный анализ его основных шагов, а также эффективность и временная сложность алгоритма. Алгоритм Кана помогает определить наличие циклов в графах, вычисляя входную степень каждого узла.

*Ключевые слова:* алгоритм Кана, топологическое упорядочение, ориентированный ациклический граф (DAG), обнаружение циклов, анализ графа, эффективность алгоритма.

*Annotation:* This paper explores Kahn's algorithm, which is widely used to determine topological order in directed acyclic graphs (DAGs). Kahn's algorithm is used to verify the acyclic structure of graphs and ensure efficient ordering of their nodes. In the article, the principle of operation of the algorithm, a detailed analysis of its main steps, as well as the efficiency and time complexity of the algorithm are considered. Kahn's algorithm helps to determine the presence of cycles in graphs by calculating the in-degree of each node.

*Key words:* Kahn's algorithm, topological ordering, directed acyclic graph (DAG), cycle detection, graph analysis, algorithm efficiency

Graflar nazariyasi kompyuter fanlarida, matematikada va axborot texnologiyalarida keng qoʻllaniladigan asosiy konseptlardan biridir. Graflar turli sohalarda, jumladan, tarmoq aloqalari, jarayonlarni boshqarish, algoritmlar va ma'lumotlar strukturalarini tasvirlashda muhim rol oʻynaydi. Ayniqsa, yoʻnalgan atsiklik graflar (DAG — Directed Acyclic Graphs) koʻplab muammolarni model qilishda va yechishda samarali vositalardan hisoblanadi. DAGlar, sikl mavjud boʻlmagan va bir yoʻnalishda rivojlanadigan graflardir. Ularning asosiy xususiyati — har bir tugun oʻzidan oldingi tugunlarga qarab tartiblanadi, bu esa ularni topologik tartibda joylashtirishni taqozo qiladi.

**Topologik tartiblash** — bu grafdagi tugunlarni, ularning bogʻliqliklaridan kelib chiqqan holda, ma'lum bir tartibda joylashtirish jarayonidir. Bu jarayonning asosiy maqsadi — har bir tugun, uning barcha kirish qirralarini olib boʻlganidan keyin ishlov berilishi kerak boʻlgan tartibni topishdir. Yoʻnalgan graflarda topologik tartibni olishning turli usullari mavjud. Ushbu maqolada, eng samarali va keng qoʻllaniladigan algoritmlardan biri — Kahn algoritmi (Kahn's Algorithm) koʻrib chiqiladi. Kahn algoritmi, yoʻnalgan atsiklik grafdagi tugunlarni topologik tartibda joylashtirish uchun ishlatiladi va agar grafdagi sikl mavjud boʻlsa, uni aniqlashga yordam beradi. Algoritmning asosi — har bir

tugunning kirish darajasini (in-degree) hisoblash va in-degree = 0 bo'lgan tugunlarni ishlov berishdan boshlash. Bu jarayon, siklni aniqlash va tugunlar o'rtasidagi tartibni to'g'ri tashkil etish imkonini beradi.

**Muhokama va natijalar:**

**Muhokama:** Kahn algoritmi, yo'nalgan atsiklik graflarda (DAG) topologik tartibni aniqlash va siklni tekshirishda samarali usul sifatida tanilgan. Maqolada taqdim etilgan tahlil va misollar orqali, algoritmning ishlash printsipi, uning samaradorligi va cheklovlari aniq ko'rsatilgan. Kahn algoritmi grafdagi har bir tugunning kirish darajasini hisoblash orqali topologik tartibni olishni ta'minlaydi. Algoritmning asosiy afzalligi shundaki, u siklni aniqlash va topologik tartibni olishni bitta jarayonda amalga oshiradi. Bu, uni ko'plab dasturiy ilovalar va algoritmik muammolar uchun juda qulay qiladi, chunki ko'plab tizimlar va jarayonlar siklsiz tuzilishga asoslangan.

Kahn algoritmining samaradorligini tahlil qilishda, uning vaqt murakkabligi  $O(V + E)$  ga tengligi alohida ahamiyatga ega. Bu shuni anglatadiki, algoritmning ishlash muddati graflarning tugunlari (V) va qirralari (E) soniga bog'liq bo'lib, bu uning katta graflarda ham samarali ishlashini ta'minlaydi. Shuningdek, Kahn algoritmi graflarda sikl mavjudligini aniqlashda ham juda foydalidir, chunki agar grafda sikl bo'lsa, topologik tartibni olish imkonsiz bo'ladi. Bu algoritmda siklni aniqlashning o'ziga xos jihati, agar algoritm grafni to'liq o'rganib chiqib, barcha tugunlar uchun topologik tartibni yaratolmasa, unda sikl mavjudligi haqida to'g'ri xulosa chiqaradi.

Biroq, Kahn algoritmidam ham ba'zi cheklovlar mavjud. U faqat atsiklik yo'nalgan graflar bilan ishlaydi, ya'ni agar grafda sikl bo'lsa, algoritm ishlamaydi. Shuningdek, juda katta o'lchamdagi graflar bilan ishlashda, xotira va resurslar bilan bog'liq muammolar yuzaga kelishi mumkin. Garchi algoritmning samaradorligi yuqori bo'lsa-da, ba'zi holatlarda boshqa topologik tartib algoritmlaridan, masalan, Depth-First Search (DFS) asosidagi usullardan ko'ra ko'proq xotira talab qilishi mumkin.

**Masala:**

Berilgan yo‘nalgan graflarda topologik tartibni aniqlash. Agar grafda sikl mavjud bo‘lsa, topologik tartibni qaytarib bo‘lmaydi.

**Dasturda quyidagilar amalga oshiriladi:**

1. Graflarni tugunlar va qirralar bilan tasvirlash.
2. Kahn algoritmi yordamida topologik tartibni olish.
3. Agar grafda sikl bo‘lsa, xato xabarini chiqarish.

**C# Dasturi:**

```
using System;
using System.Collections.Generic;
class Graph
{ private int V; // Grafdagi tugunlar soni
  private List<int>[] adjList; // Grafni tasvirlovchi qo'shma ro'yxat
  // Konstruktor
  public Graph(int vertices)
  { V = vertices; adjList = new List<int>[V]; for (int i = 0; i < V; i++)
adjList[i] = new List<int>(); }
  // Grafga yo'nalgan qirrani qo'shish
  public void AddEdge(int u, int v)
  { adjList[u].Add(v); }
  // Kahn algoritmi yordamida topologik tartibni aniqlash
  public bool TopologicalSort()
  { int[] inDegree = new int[V]; // Har bir tugunning kirish darajasini
saqlashQueue<int> queue = new Queue<int>();
  // Har bir tugunning kirish darajasini hisoblash
  for (int i = 0; i < V; i++)
  { foreach (var neighbor in adjList[i]) inDegree[neighbor]++; } //
Kiritish darajasi 0 bo'lgan tugunlarni navbatga qo'shish
  for (int i = 0; i < V; i++) {
    if (inDegree[i] == 0) queue.Enqueue(i); }
```

```

List<int> topologicalOrder = new List<int>(); // Topologik tartibni
saqlash while (queue.Count > 0)
{
    int current = queue.Dequeue(); topologicalOrder.Add(current);
    // Uskunadagi barcha qo'shni tugunlar uchun kirish darajasini kamaytirish
    foreach (var neighbor in adjList[current]){inDegree[neighbor]--;if
(inDegree[neighbor] == 0)queue.Enqueue(neighbor);}}// Agar topologik tartib
barcha tugunlarni o'z ichiga olsaif (topologicalOrder.Count !=
V){Console.WriteLine("Grafda sikl mavjud!");return false; // Sikl bor, topologik
tartibni olish imkonsiz}// Topologik tartibni
chiqarishConsole.WriteLine("Topologik Tartib: ");foreach (var node in
topologicalOrder){Console.Write(node + " ");}Console.WriteLine();return
true;}}class Program{static void Main(string[] args){// Grafni yaratish (masalan,
6 ta tugun)Graph g = new Graph(6);
    // Yo'nalgan qirralarni qo'shish
    g.AddEdge(5, 2);
    g.AddEdge(5, 0);
    g.AddEdge(4, 0);
    g.AddEdge(4, 1);
    g.AddEdge(2, 3);
    g.AddEdge(3, 1);
    // Topologik tartibni olish
    bool result = g.TopologicalSort();
    // Agar natija topologik tartib bo'lsa, programmaning yakuniy xabariif
(result){Console.WriteLine("Topologik tartib muvaffaqiyatli olishdi.");}else{
Console.WriteLine("Topologik tartibni olishning iloji yo'q.");}
Console.ReadLine();}}

```

### Dastur ishlash printsiipi:

1. **Graph** klassi grafni ifodalaydi va uning tugunlari o'rtasida yo'nalgan qirralar qo'shadi.
2. **TopologicalSort** metodi Kahn algoritmini amalga oshiradi:

- Har bir tugunning kirish darajasini (in-degree) hisoblaydi.
  - In-degree = 0 bo'lgan tugunlar navbatga qo'yiladi.
  - Navbatdagi tugunlardan chiqish qirralari olib tashlanadi va ularning kirish darajalari kamaytiriladi.
  - Agar barcha tugunlar o'rganilsa, topologik tartib qaytariladi.
  - Agar ba'zi tugunlar ishlanmagan bo'lsa, sikl mavjudligi aniqlanadi.
3. Main metodida graf yaratiladi, unga qirralar qo'shiladi va TopologicalSort metodi chaqiriladi.

### **Natija:**

Ushbu dastur Topologik Tartib ni konsolga chiqaradi, agar graf siklsiz bo'lsa. Agar grafda sikl mavjud bo'lsa, "Grafda sikl mavjud!" deb xabar beradi.

Ushbu maqolada, Kahn algoritmi yordamida yo'nalgan atsiklik graflarda (DAG) topologik tartibni aniqlash masalasi ko'rib chiqildi. Kahn algoritmi — graflarda siklni aniqlash va topologik tartibni olishda samarali va keng qo'llaniladigan metodlardan biridir. Maqolada algoritmning ishlash printsipi, uning afzalliklari va cheklovlari, shuningdek, C# dasturi orqali uning amaliy qo'llanilishi ko'rsatildi. Dasturda Kahn algoritmi C# dasturlash tilida amalga oshirildi va uni ishlatish uchun graflarga qirralar qo'shib, topologik tartib olish jarayoni ko'rsatilgan. Dasturda agar grafda sikl bo'lsa, "Grafda sikl mavjud!" deb xabar beriladi. Agar sikl mavjud bo'lmasa, topologik tartib konsolga chiqariladi.

### **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. 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.
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.