

TOPOLOGIK TARTIBLASH: KAHN ALGORITMINING TEZKOR YONDASHUVI

Farmonov Sherzod Raxmonjonovich

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

[*farmonovsh@gmail.com*](mailto:farmonovsh@gmail.com)

A'zamova Sevinch Umidjon qizi

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

[*dilafruzahmadjonova4@gmail.com*](mailto:dilafruzahmadjonova4@gmail.com)

Annotatsiya: Maqolada, "Topologik Tartiblash: Kahn algoritmining tezkor yondashuvi", topologik tartiblash masalasini yechishda Kahn algoritmining samarali qo'llanilishi va uning tezkor yondashuvi muhokama qilinadi. Kahn algoritmi, yo'llar orasidagi bog'lanishlar (direkt grafda) orqali tsikl mavjud bo'lmagan elementlarning tartiblanishini ta'minlashga mo'ljallangan bo'lib, u $O(n + m)$ vaqt kompleksligi bilan ishlaydi, bu yerda n — tugunlar soni, m — qirralar soni. Maqola, Kahn algoritmiga asoslangan yondashuvning samaradorligini tahlil qiladi, ayniqsa yirik va murakkab graflarda.

Kalit so'zlar: Topologik tartiblash, Kahn algoritmi, Tezkor yondashuv, Direkt graf, Tsiklsiz graf, Vaqt kompleksligi, Algoritm optimallashtirish, Parallel hisoblash, Graflar, Dasturiy ta'minot, Tizimlar nazariyasi, Amaliy qo'llanilish, Yirik graf, Tezkor algoritmlar

Abstract: The paper, "Topological Sorting: A Fast Approach to Kahn's Algorithm", discusses the effective application of Kahn's algorithm and its fast approach to solving the topological sorting problem. Kahn's algorithm is designed to ensure the ordering of non-cyclic elements through the links between paths (in a directed graph), and it works with $O(n + m)$ time complexity, where n is the number of nodes, m is the number of edges. The article analyzes the efficiency of the approach based on the Kahn algorithm, especially on large and complex graphs.

Keywords: Topological Sorting, Kahn Algorithm, Fast Approach, Directed Graph, Acyclic Graph, Time Complexity, Algorithm Optimization, Parallel Computing, Graphs, Software, Systems Theory, Practical Application, Large Graph, Fast Algorithms

Аннотация: В статье «Топологическая сортировка: быстрый подход к алгоритму Кана» обсуждается эффективное применение алгоритма Кана и его быстрый подход к решению задачи топологической сортировки. Алгоритм Кана предназначен для обеспечения упорядочивания нециклических элементов

посредством связей между путями (в ориентированном графе) и работает с временной сложностью $O(n + m)$, где n — количество узлов, m — число краев. В статье анализируется эффективность подхода на основе алгоритма Кана, особенно на больших и сложных графах.

Ключевые слова: топологическая сортировка, алгоритм Кана, быстрый подход, ориентированный граф, ациклический граф, временная сложность, оптимизация алгоритма, параллельные вычисления, графы, программное обеспечение, теория систем, практическое применение, большой граф, быстрые алгоритмы.

Kirish. Topologik tartiblash, graf nazariyasining muhim va keng qo'llaniladigan masalalaridan biri bo'lib, u asosan tsiklsiz yo'llar orasidagi tartibni aniqlashni maqsad qiladi. Bu masala, ko'plab sohalarda, jumladan, dasturiy ta'minot ishlab chiqish, vaqtni boshqarish tizimlari, resurslarni taqsimlash, va tarmoqlarni optimallashtirishda dolzarb ahamiyatga ega. Topologik tartiblashning asosiy maqsadi — berilgan grafning barcha tugunlarini, ularning qirralari bo'yicha, ma'lum bir tartibda joylashtirishdir, bunda har bir tugun faqat unga kiruvchi tugunlar tartibida joylashadi.

Kahn algoritmi — topologik tartiblashni yechishda ishlatiladigan eng mashhur metodlardan biri bo'lib, unda grafning tarkibiy qismlarini bosqichma-bosqich qayta ishlash orqali, tsiklsiz grafda tartibni aniqlash mumkin. Algoritmning afzalliklaridan biri uning oddiyliги va samaradorligidir. Biroq, yirik va murakkab graf strukturalarida uning ishlash samaradorligini oshirish uchun optimallashtirishlar va parallel hisoblash yondashuvlari zarur bo'ladi.

Muhokama va natijalar:

Kahn algoritmi, topologik tartiblash masalasini yechishda samarali va intuitiv yondashuvni taqdim etadi. Uning asosiy printsiپی — grafikning barcha tugunlarini, ular orasidagi qirralar asosida, qaysi tugunlar avvalgi tugunlarga bog'langanini inobatga olib, tartibga solishdir. Algoritmning ishlash jarayoni nisbatan oddiy: har bir tugunning kiruvchi qirralarini hisobga olib, kiruvchi qirralari bo'lmagan tugunlarni chiqish ro'yxatiga qo'shib boradi, bu jarayon davomida boshqa tugunlarga nisbatan bog'liqlikni kuzatib boradi. Ushbu metodning samaradorligi, algoritmning $O(n + m)$ vaqt kompleksligi bilan ishlashda yotadi, bu n — tugunlar soni, m — qirralar soni bo'lib, bu murakkablik eng yaxshi holatlardan biridir.

Biroq, Kahn algoritmining standart versiyasi katta hajmdagi graflarda samaradorlik nuqtai nazaridan ba'zi cheklovlarga ega bo'lishi mumkin. Ayniqsa, yirik tarmoqlarda va ko'plab bog'lanishlarga ega bo'lgan graflarda, algoritmning serial ishlash usuli ko'p vaqt va resurs talab qilishi mumkin. Shuning uchun, maqolada Kahn

algoritmining parallel ishlash variantlari va ularni optimallashtirishga oid yondashuvlar ham ko'rib chiqiladi. Parallelizatsiya yordamida, grafni bir nechta bo'laklarga bo'lish va har bir bo'lakni alohida qayta ishlash orqali, topologik tartiblash jarayonining umumiy vaqtini sezilarli darajada qisqartirish mumkin. Bu yondashuvlar, ayniqsa, bulutli hisoblash tizimlari yoki katta hajmdagi ma'lumotlar bilan ishlashda samarali bo'lishi mumkin.

Shuningdek, maqolada topologik tartiblashning boshqa sohalarda, xususan, dasturiy ta'minot ishlab chiqish, resurslarni taqsimlash va tarmoqlarni optimallashtirish kabi amaliy qo'llanilishlari tahlil qilinadi. Misol uchun, dasturiy ta'minotdagi modul o'rtasidagi bog'lanishlarni aniqlashda, topologik tartiblashning yordamida modular tizimlarning to'g'ri yuklanish tartibi belgilanishi mumkin. Tizimlarni tahlil qilish va optimallashtirishda esa, resurslarni taqsimlash va ustuvorlikni belgilashda ushbu algoritm samarali vosita sifatida foydalaniladi.

Natijalar:

1. **Kahn algoritmining samaradorligi:** Kahn algoritmi, tsiklsiz graflarda topologik tartiblashni samarali yechishga imkon beradi. Uning vaqt kompleksligi $O(n + m)$ bo'lib, bu katta graflarda ham yaxshi ishlashini ta'minlaydi. Biroq, yirik graflarda uning samaradorligini oshirish uchun parallel yondashuvlar kerak bo'lishi mumkin.

2. **Amaliy qo'llanilishlar:** Topologik tartiblash algoritmlari dasturiy ta'minot dizaynida, tizimlar nazariyasida, tarmoqda resurslarni taqsimlashda keng qo'llaniladi. Kahn algoritmi o'zining oddiyliги va samaradorligi tufayli, bu sohalarda amaliy yechimlarni taqdim etishga imkon beradi.

3. **Kelajakdagi tadqiqot yo'nalishlari:** Kahn algoritmining parallel ishlash variantlarini yanada takomillashtirish va uni murakkab tizimlarda qo'llashda yangi yondashuvlar izlanishi zarur. Shuningdek, graflarda tsikllarni aniqlash va tsikl bo'lgan holatlarni samarali ravishda boshqarish muammolarini hal qilishga oid tadqiqotlar ham jadal davom etmoqda.

Masala: Topologik Tartiblash

Agar sizda quyidagi yo'nalgan graf bo'lsa:

- $A \rightarrow B$
- $A \rightarrow C$
- $B \rightarrow D$
- $C \rightarrow D$
- $D \rightarrow E$

Grafdagi tugunlar: A, B, C, D, E.

Sizga kerak:

1. Ushbu grafni Kahn algoritmi yordamida topologik tartibda tartiblash.
2. Kahn algoritmining ishlash jarayonini batafsil ko'rsatish.

Kahn algoritmini qo'llash:

Qadam 1: Dastlab, har bir tugun uchun uning kirituvchi (incoming) cheklovlarini sanab chiqing:

- A: 0 (hech kim bu tugunga kiritish qilmaydi)
- B: 1 (faqat A kiritadi)
- C: 1 (faqat A kiritadi)
- D: 2 (B va C kiritadi)
- E: 1 (D kiritadi)

Qadam 2: Kirituvchi cheklovlari 0 bo'lgan tugunlarni qatorga qo'shing. Bu holda A tuguni kirituvchi cheklovlari 0 ga teng, shuning uchun uni boshlash nuqtasi sifatida qatorga qo'shamiz.

Qadam 3: A tuguni qatorga qo'shilgandan so'ng, uning chiqishlaridagi (B va C) kirituvchi cheklovlarini kamaytiring:

- B: $1 \rightarrow 0$
- C: $1 \rightarrow 0$

Endi B va C ham kirituvchi cheklovlari 0 bo'lgani uchun, ularni ham qatorga qo'shamiz.

Qadam 4: B va C tugunlari qatorga qo'shilgandan so'ng, D tugunining kirituvchi cheklovlari 2 bo'lishi kerak (B va C tomonidan). Ammo endi B va C tugunlarining kirituvchi cheklovlari 0 bo'lganligi uchun, D tugunining kirituvchi cheklovlari $2 \rightarrow 0$ bo'ladi va D tuguni qatorga qo'shiladi.

Qadam 5: D tuguni qatorga qo'shilgandan so'ng, E tugunining kirituvchi cheklovlarini kamaytiramiz:

- E: $1 \rightarrow 0$

Endi E tugunini qatorga qo'shamiz.

Qadam 6: Endi barcha tugunlar qatorga qo'shildi, va bizning topologik tartibimiz quyidagicha bo'ladi:

Topologik Tartib: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

Kahn algoritmining ishlash jarayoni:

1. **A** tuguni kirituvchi cheklovlari 0 bo'lganligi uchun birinchi bo'lib qatorga qo'shildi.
2. **B** va **C** tugunlari **A** tugunidan kiritilganligi uchun ularning kirituvchi cheklovlari 0 ga tushdi va ular qatorga qo'shildi.
3. **D** tuguni **B** va **C** tugunlaridan kiritilganligi uchun ularning kirituvchi cheklovlari 0 ga tushganidan so'ng, **D** tuguni qatorga qo'shildi.
4. **E** tuguni **D** tugunidan kiritilganligi uchun, **D** tuguni qatorga qo'shilgandan so'ng, **E** tuguni ham qatorga qo'shildi.

Natija:

Topologik tartiblashning natijasi: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

Quyida C# dasturlash tilida Kahn algoritmi yordamida berilgan topologik tartiblash masalasini qanday yechishni ko'rsatib beradi.

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

class Program
{
    static void Main()
    {
        // Grafni aniqlash
        var graph = new Dictionary<string, List<string>>()
        {
            { "A", new List<string> { "B", "C" } },
            { "B", new List<string> { "D" } },
            { "C", new List<string> { "D" } },
            { "D", new List<string> { "E" } },
            { "E", new List<string>() }
        };

        // Topologik tartibni topish
```

```

var topologicalOrder = TopologicalSort(graph);

// Natijani chiqarish
if (topologicalOrder != null)
{
    Console.WriteLine("Topologik tartib: ");
    foreach (var node in topologicalOrder)
    {
        Console.Write(node + " ");
    }
}
else
{
    Console.WriteLine("Topologik tartibni topish imkonsiz, graf tsiklga
ega.");
}
}

// Kahn algoritmi bilan topologik tartiblash
static List<string> TopologicalSort(Dictionary<string, List<string>> graph)
{
    var inDegree = new Dictionary<string, int>();
    var queue = new Queue<string>();
    var topologicalOrder = new List<string>();

    // Har bir tugun uchun kirituvchi cheklovlarni (in-degree) hisoblash
    foreach (var node in graph.Keys)
    {
        inDegree[node] = 0; // Avval kirituvchi cheklovlarni 0 ga tenglaymiz
    }

    foreach (var node in graph)
    {
        foreach (var neighbor in node.Value)
        {
            inDegree[neighbor]++;
        }
    }
}

```

```

// Kirituvchi cheklovlari 0 bo'lgan tugunlarni boshlang'ich qatorga qo'shamiz
foreach (var node in inDegree)
{
    if (node.Value == 0)
    {
        queue.Enqueue(node.Key);
    }
}

// Kahn algoritmi
while (queue.Count > 0)
{
    var currentNode = queue.Dequeue();
    topologicalOrder.Add(currentNode);

    // Hozirgi tugundan chiqqan barcha tugunlar uchun kirituvchi cheklovlarni
kamaytiramiz
    if (graph.ContainsKey(currentNode))
    {
        foreach (var neighbor in graph[currentNode])
        {
            inDegree[neighbor]--;
            if (inDegree[neighbor] == 0)
            {
                queue.Enqueue(neighbor);
            }
        }
    }
}

// Agar topologik tartibda barcha tugunlar mavjud bo'lsa, natijani qaytaramiz
if (topologicalOrder.Count == graph.Keys.Count)
{
    return topologicalOrder;
}
else
{
    // Agar graf tsiklga ega bo'lsa, topologik tartibni topish imkonsiz
    return null;
}

```

```

    }
  }
}

```

Chiqarish:

Agar yuqoridagi kodni ishlatsangiz, quyidagi natijani ko'rasiz:

Topologik tartib:

A B C D E

Xulosa:

Bu maqola, Kahn algoritmining samarali ishlash printsipi, C# dasturlash tilida qanday amalga oshirilishini tushuntiradi va topologik tartiblashning amaliy ahamiyatini ko'rsatadi. Bu algoritmni tushunish va ishlatish ko'plab sohalarda, masalan, dasturiy ta'minot tizimlari, tarmoq modellashtirish, va graf nazariyasida foydalidir.

Foydalanilgan adabiyotlar:

1. Kahn, A. B. (1962). "Topological sorting of large networks". *Communications of the ACM*, 5(11), 558-562.
2. Sedgewick, R., & Wayne, K. (2011). *Algorithms (4th ed.)*. Addison-Wesley.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms (3rd ed.)*. MIT Press.
4. Knuth, D. E. (1997). *The Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd ed.)*. Addison-Wesley.
5. Shen, H., & Wang, J. (2013). *Graph Theory and Complex Networks: An Introduction*. Springer.
6. Microsoft Documentation (2023). *C# Programming Guide*.