KOSARADJA ALGORITMI

Onarkulov Maksadjon Karimberdiyevich
Farg'ona davlat univarsiteti Amaliy matematika va informatika kafedrasi dotsenti (PhD)
maxmaqsad@gmail.com
Soliyeva Xurshida Tavakaljon qizi
Farg'ona davlat universiteti talabasi
e-mail: xurshidasoliyeva56@gmail.com
Qodirova Gulnoraxon Akmaljon qizi
Farg'ona davlat universiteti talabasi
qodirovag035@gmail.com

ANNOTATSIYA. *Ushbu maqolada Kosaraju algoritmi va uning yo'naltirilgan grafikdagi kuchli bog'langan komponentalarni aniqlashdagi ahamiyati ko'rib chiqiladi. Maqolada algoritmning ishlash bosqichlari, vaqt murakkabligi, qo'llanilish sohalari batafsil tahlil qilinadi. Shuningdek, C# dasturlash tilida yozilgan dasturiy realizatsiyasi ham keltirilgan. Bu algoritm ma'lumotlarni tahlil qilish, ijtimoiy tarmoqlar va kompyuter tarmoqlarini tahlil qilish kabi sohalarda keng qo'.llaniladi.*

Kalit so'zlar: *Kosaraju algoritmi, yo'naltirilgan graf, kuchli bog'langan komponenta, grafik algoritmlar, DFS (chuqurlik bo'yicha qidiruv), orqaga qaytish, grafik tahlili, algoritm murakkabligi, C# dasturlash tili.*

ANNOTATION. *This article explores the Kosaraju algorithm and its importance in detecting strongly connected components in a directed graph. The steps of the algorithm, time complexity, and its applications are discussed in detail. A practical implementation of the algorithm in the C# programming language is also presented. The algorithm is widely used in data analysis, social network analysis, and computer network systems.*

Keywords: *Kosaraju algorithm, directed graph, strongly connected components, graph algorithms, DFS (depth-first search), backtracking, graph analysis, algorithm complexity, C# programming language.*

АННОТАЦИЯ *. В данной статье рассматривается алгоритм Косараджа и его значение для нахождения сильно связных компонентов в ориентированном графе. Подробно описываются этапы выполнения алгоритма, его временная сложность и области применения. Также представлена реализация алгоритма на языке программирования C#. Этот алгоритм широко используется в анализе данных, социальных сетей и компьютерных сетей.*

Ключевые слова: *Алгоритм Косараджа, ориентированный граф, сильно связные компоненты, алгоритмы графов, поиск в глубину (DFS), обратный ход, анализ графов, сложность алгоритма, язык программирования C#.*

Kirish.

Hozirgi zamonaviy axborot texnologiyalari davrida grafiklar asosidagi masalalarni hal qilish algoritmlari juda katta ahamiyat kasb etadi. Xususan, yo‘naltirilgan grafdagi kuchli bog‘langan komponentalarni aniqlash – kompyuter tarmoqlari, ijtimoiy tarmoq analizlari, dasturiy ta’minotlar strukturasi va boshqa ko‘plab muhim sohalarda qo‘llaniladi. Bunday graf strukturalarini tahlil qilishda samarali algoritmlar zarur bo‘ladi. Shunday algoritmlardan biri — **Kosaraju algoritmi** hisoblanadi.Kosaraju algoritmi Hindistonlik informatika olimi **S.R. Kosaraju** tomonidan ishlab chiqilgan va u **yo‘naltirilgan graf** ichidagi barcha **kuchli bog‘langan komponentalarni (Strongly Connected Components, SCCs)** aniqlashda foydalaniladi. Ushbu algoritm o‘zining soddaligi, tushunarli bosqichlari va samaradorligi bilan boshqa algoritmlardan (masalan, Tarjan yoki Gabow algoritmlaridan) ajralib turadi.Kosaraju algoritmi ikkita asosiy DFS (Depth-First Search – chuqurlik bo‘yicha qidiruv) chaqirig‘i asosida ishlaydi.

Birinchi DFS graf ustida o‘tkaziladi va tugallanish vaqtlari asosida tugunlar tartiblanadi. Ikkinchi DFS esa grafning teskari ko‘rinishida amalga oshiriladi va aynan shu bosqichda kuchli bog‘langan komponentalar aniqlanadi.Bu algoritm **O(N + M)** (bu yerda N – tugunlar soni, M – qirralar soni) vaqt murakkabligiga ega bo‘lib, katta o‘lchamli grafiklarda ham samarali ishlaydi. Shu sababli, Kosaraju algoritmi nafaqat nazariy jihatdan, balki amaliy dasturlarda ham keng qo‘llaniladi.Mazkur maqolada Kosaraju algoritmining ishlash tamoyillari, uning afzalliklari va C# dasturlash tilidagi dasturiy ko‘rinishi to‘liq yoritib beriladi. Bundan tashqari, kuchli bog‘langan komponentalarni aniqlashga oid misollar, kodlar va algoritmning harakat bosqichlari ham batafsil tushuntiriladi.

Asosiy qism

Kosaraju algoritmi yo‘naltirilgan grafdagi barcha kuchli bog‘langan komponentalarni aniqlash uchun qo‘llaniladigan samarali algoritmdir Har bir kuchli bog‘langan komponenta bu grafikda shunday tugunlar to‘plamiki ulardan har qanday biridan boshqasiga va aksincha yo‘l mavjud

Kosaraju algoritmi uchta asosiy bosqichda amalga oshiriladi

Birinchi bosqichda grafik ustida chuqurlik bo‘yicha qidiruv DFS bajariladi Har bir tugundan DFS chaqiriladi va tugunlarning tugallanish vaqtlari stackga joylanadi

Ikkinchi bosqichda grafikning teskari ko‘rinishi olinadi ya’ni barcha yo‘nalishlar teskari qilinadi Masalan A dan B ga yo‘nalish bo‘lsa u B dan A ga aylantiriladi

Uchinchi bosqichda esa tugunlar birinchi bosqichdagi DFS tugallanish tartibiga ko‘ra stackdan olinadi va har bir tugun uchun teskari grafda yana DFS chaqiriladi Har bir DFS chaqiruvi yangi kuchli bog‘langan komponentani hosil qiladi

Kosaraju algoritmining umumiy vaqt murakkabligi O(N + M) bo‘lib bu yerda N tugunlar soni M esa qirralar sonini anglatadi Bu murakkablik uni katta grafiklarda ham samarali ishlatish imkonini beradi

Ushbu algoritmni C# dasturlash tilida quyidagicha amalga oshirish mumkin

csharp

Копировать код

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

class Kosaraju
{
    static int vertices;
    static List<int>[] graph;
    static List<int>[] reverseGraph;
    static bool[] visited;
    static Stack<int> stack = new Stack<int>();

    static void Main()
    {
        vertices = 5;

        graph = new List<int>[vertices];
        reverseGraph = new List<int>[vertices];
        for (int i = 0; i < vertices; i++)
        {
            graph[i] = new List<int>();
            reverseGraph[i] = new List<int>();
        }
```

```
    AddEdge(0, 2);
    AddEdge(2, 1);
    AddEdge(1, 0);
    AddEdge(0, 3);
    AddEdge(3, 4);

    visited = new bool[vertices];
    for (int i = 0; i < vertices; i++)
      if (!visited[i])
        DFS1(i);

    for (int i = 0; i < vertices; i++)
      foreach (var v in graph[i])
        reverseGraph[v].Add(i);

    visited = new bool[vertices];
    Console.WriteLine("Kuchli bog‘langan komponentalar:");
    while (stack.Count > 0)
    {
      int v = stack.Pop();
      if (!visited[v])
      {
        DFS2(v);
        Console.WriteLine();
      }
    }
}

static void AddEdge(int u, int v)
{
```

```
        graph[u].Add(v);
    }

    static void DFS1(int v)
    {
        visited[v] = true;
        foreach (int u in graph[v])
            if (!visited[u])
                DFS1(u);
        stack.Push(v);
    }

    static void DFS2(int v)
    {
        visited[v] = true;
        Console.Write(v + " ");
        foreach (int u in reverseGraph[v])
            if (!visited[u])
                DFS2(u);
    }
}
```

Ushbu dastur orqali grafikdagi barcha kuchli bog‘langan komponentalar aniqlanadi Har bir komponenta yangi satrda chiqariladi va bu grafik strukturani tahlil qilishda juda qo‘l keladi

Xulosa

Kosaraju algoritmi yo‘naltirilgan grafiklarda kuchli bog‘langan komponentalarni aniqlashda samarali va ishonchli algoritm hisoblanadi Ushbu algoritmning asosi bo‘lgan chuqurlik bo‘yicha qidiruv va grafni teskari aylantirish amallari orqali grafik tuzilmasidagi ichki bog‘liqliklar aniq ko‘rinadi. Algoritmning O(N + M) murakkabligi uni amaliyotda katta grafiklar bilan

ishlashda qo‘llash imkonini beradi Dasturiy ta’minot tizimlarida ijtimoiy tarmoqlarda ma’lumotlar oqimida va yo‘nalishli aloqalarni tahlil qilishda Kosaraju algoritmi keng foydalaniladi. Mazkur maqolada ushbu algoritmning ishlash mexanizmi C# dasturlash tilida aniq misollar bilan tushuntirildi Bu esa talabalar va dasturchilar uchun nazariy bilimlarni amaliy ko‘nikmalar bilan boyitishga xizmat qiladi. Kosaraju algoritmi orqali graf tahlilida samarali yondashuv shakllantiriladi va bu turdagi algoritmlar boshqa murakkab muammolarni yechishda asos bo‘lib xizmat qilishi mumkin

Foydalanilgan adabiyotlar

1. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. — Algorithms: Introduction to Algorithms, Third Edition, MIT Press, 2009
2. Sedgewick R., Wayne K. — Algorithms (4th Edition), Addison-Wesley, 2011
3. Dasgupta S., Papadimitriou C., Vazirani U. — Algorithms, McGraw-Hill, 2008
4. Skiena S. S. — The Algorithm Design Manual, Springer, 2020
5. Bondy J. A., Murty U. S. R. — Graph Theory, Springer, 2008
6. Gross J. L., Yellen J. — Graph Theory and Its Applications, CRC Press, 2018
7. Narasimha Karumanchi — Data Structures and Algorithms Made Easy, CareerMonk, 2016
8. Knuth D. E. — The Art of Computer Programming. Volume 1–4, Addison-Wesley, 2011
9. Laaksonen A. — Guide to Competitive Programming, Springer, 2020
10. Microsoft Docs – C# Programming Guide – https://learn.microsoft.com/en-us/dotnet/csharp/
11. GeeksforGeeks – Kosaraju's Algorithm for Strongly Connected Components – https://www.geeksforgeeks.org/

12. HackerRank – Graph Algorithms – https://www.hackerrank.com/

13. Leetcode Graph Problems – https://leetcode.com/problemset/algorithms/

14. Rosetta Code – Kosaraju Algorithm Implementations – https://rosettacode.org/wiki/Kosaraju%27s_algorithm

15. Neso Academy – Graph Theory Tutorials – https://www.youtube.com/@NesoAcademy