GRAF NAZARIYASIDA MINIMAL BOG'LANISH DARAXTINI TOPISHDA PRIM ALGORITMINING QO'LLANILISHI

Farmonov Sherzodbek Raxmonjonovich
Farg'ona davlat unversiteti amaliy matematika va informatika kafedrasi katta o'qituvchisi
Farmonovsh@gmail.com
Xoshimjonova Sevaraxon Elmurod qizi
Farg'ona davlat universiteti 2-kurs talabasi
xoshimjonovasevara@gmail.com

Annotatsiya: Ushbu maqolada Prim algoritmi, graf nazariyasida *minimum spanning tree (MST) topish uchun ishlatiladigan samarali algoritm sifatida ko'rib chiqiladi. Prim algoritmi, berilgan grafdagi eng kichik og'irlikka ega bo'lgan bog'lanishlarni tanlab, birinchi nuqtadan boshlanib, bosqichma-bosqich yangi nuqtalarni qo'shish orqali MST ni quradi. Maqolada algoritmning ishlash prinsipi, uning murakkabligi, shuningdek, amaliyotda qo'llanilishi va boshqa MST algoritmlari bilan taqqoslanishi batafsil bayon etiladi. Shuningdek, Prim algoritmining turli xil variantlari va ularning afzalliklari ham ko'rib chiqiladi.*

Kalit so'zlar: *Prim algoritmi, Minimum spanning tree(MST), Graf nazariyasi,algoritm murakkabligi, amaliy qo'llanish, MST algoritmlari,bog'lanishlar,og'irlik,turli xil variantlar*

Annotation: *This article examines Prim's algorithm as an efficient method for finding the minimum spanning tree (MST) in graph theory. Prim's algorithm selects the smallest weight connections in a given graph, starting from an initial point and incrementally adding new points to construct the MST. The article details the working principle of the algorithm, its complexity, as well as its practical applications and comparisons with other MST algorithms. Additionally, various variants of Prim's algorithm and their advantages are discussed.*

ISSN 3060-4567

Keywords: Prim's algorithm, Minimum spanning tree (MST), Graph theory, Algorithm complexity, Practical applications, MST algorithms, Connections, Weight, Various variants

Аннотация:*В данной статье рассматривается алгоритм Прима как эффективный алгоритм для поиска минимального остовного дерева (MST) в теории графов. Алгоритм Прима строит MST, выбирая соединения с наименьшим весом, начиная с одной вершины и постепенно добавляя новые вершины. В статье подробно описываются принцип работы алгоритма, его сложность, а также применение на практике и сравнение с другими алгоритмами MST. Также рассматриваются различные варианты алгоритма Прима и их преимущества.*

Ключевые слова: *Алгоритм Прима, Минимальное остовное дерево (MST), Теория графов, Сложность алгоритма, Практическое применение, Алгоритмы MST, Соединения, Вес, Различные варианты*

Prim algoritmi – bu graf nazariyasida minimal ostov daraxtini (MST) topish uchun ishlatiladigan samarali algoritm. Ushbu algoritm, berilgan bog'lanishlar (graf) ichidan eng kichik og'irlikka ega bo'lgan ostov daraxtini qurishga qaratilgan.

Qo'llanilishi: Prim algoritmi ko'plab amaliy masalalarda, masalan, tarmoq dizayni, yo'l qurilishi va boshqa sohalarda qo'llaniladi. U asosan bog'lanishlarning og'irliklari bilan ishlaydigan masalalarda samarali hisoblanadi. Shuningdek, Prim algoritmi, Kruskal algoritmi bilan birga, minimal ostov daraxtini topishning eng mashhur usullaridan biri hisoblanadi.

Prim algoritmining bosqichlari

Prim algoritmi minimal ostov daraxtini (MST) topish uchun quyidagi bosqichlarda amalga oshiriladi:

1.Boshlanish:

Tugmalarni tanlash: Algoritmni boshlash uchun biror bir tugmani (vertex) tanlaymiz.Bu tugma MSTning boshlanish nuqtasi bo'ladi.

Ostov daraxtini yaratish: Tanlangan tugmani MSTga qo'shiladi.

2.Bog’lanishlarni kuzatish:

Qayta ko'rib chiqish: Hozirgi MSTga qo'shilgan tugmalar atrofidagi bog'lanishlarni (edges) kuzating. Bu bog'lanishlar MSTga qo'shilishi mumkin bo'lgan tugmalar bilan bog'langan bo'lishi kerak.

3.Eng kichik bog'lanishni tanlash:

Tanlov: Hozirgi MSTga qo'shilgan tugmalar atrofidagi bog'lanishlar orasidan eng kichik og'irlikka ega bo'lgan bog'lanishni tanlang. Bu bog'lanish orqali yangi tugma MSTga qo'shiladi.

4. Yangi tugmani qo'shish:

Qo'shish: Tanlangan bog'lanish orqali yangi tugmani MSTga qo'shing va uni MSTning bir qismi sifatida belgilab qo'ying.

Qo'shimcha izohlar:

• Prioritet navbatlari: Algoritm samaradorligini oshirish uchun, eng kichik bog'lanishni tezda topish uchun prioritet navbatlari (masalan, min-heap) ishlatilishi mumkin.

• Og'irliklar: Agar grafda og'irliklar berilmagan bo'lsa, u holda bog'lanishlar teng deb hisoblanadi va har qanday bog'lanish tanlanishi mumkin.

Bu bosqichlar orqali Prim algoritmi minimal ostov daraxtini samarali tarzda topadi.

Prim algoritmining qo’llanilish sohalari

Prim algoritmi, minimal ostov daraxtini (MST) topish uchun ishlatiladigan samarali algoritm bo'lib, ko'plab sohalarda qo'llaniladi. Quyida uning asosiy qo'llanilish sohalari keltirilgan:

1.Tarmoq dizayni:

• Telekommunikatsiya tarmoqlari: Telefon va internet tarmoqlarini qurishda minimal xarajatli bog'lanishlarni aniqlash uchun.

• Transport tarmoqlari: Yo'llar, temir yo'llar yoki boshqa transport infratuzilmasini loyihalashda optimal yo'nalishlarni belgilashda.

2.Ma'lumotlar tuzilishi va algoritmlar

• Grafikalar: Katta grafikalar ustida ishlashda, masalan, ijtimoiy tarmoqlar yoki ma'lumotlar bazalaridagi aloqalarni aniqlashda.

• O'qituvchi tizimlar: Ma'lumotlarni optimallashtirish va samarali strukturalarni yaratishda.

2.Kiberxavfsizlik

• Tarmoq xavfsizligi: Tarmoqdagi zaif joylarni aniqlash va xavfsizlikni ta'minlashda minimal bog'lanishlarni aniqlashda.

4. O'yin Dizayni

• O'yin xaritalari: O'yinlarda xaritalarni tuzishda yoki resurslarni optimallashtirishda minimal xarajatli yo'llarni aniqlash uchun.

Kiberxavfsizlik jarayonida Prim algoritmining qo’llanilishi

Prim algoritmi – bu graf nazariyasida minimal bog‘lanish daraxtini (MST) topish uchun ishlatiladigan bir algoritmdir. Kiberxavfsizlik jarayonida Prim algoritmining qo‘llanilishi bir qancha sohalarda amalga oshirilishi mumkin. Quyida bu algoritmning kiberxavfsizlikdagi potentsial qo‘llanilishlarini keltirib o‘tamiz:

1. Tarmoq topologiyasini optimallashtirish: Kiberxavfsizlik tarmoqlarini loyihalashda, Prim algoritmi yordamida tarmoqning minimal bog‘lanish daraxtini yaratish orqali ma'lumotlar uzatish yo‘llarini optimallashtirish mumkin. Bu, o‘z navbatida, ma'lumotlarning xavfsizligini oshiradi.

2. Xavfsizlik devorlari va qurilmalar o‘rtasidagi bog‘lanishni boshqarish: Tarmoqdagi xavfsizlik devorlari va boshqa qurilmalar o‘rtasidagi ma'lumotlar almashinuvi uchun eng samarali yo‘llarni aniqlashda Prim algoritmidan foydalanish mumkin. Bu, xavfsizlik devorlari o‘rtasidagi bog‘lanishni yanada mustahkamlashga yordam beradi.

3. Ma'lumotlarni shifrlash va uzatish: Ma'lumotlarni uzatishda, Prim algoritmi yordamida bog‘lanish yo‘llarini aniqlash orqali shifrlangan ma'lumotlarning samarali va xavfsiz uzatilishini ta'minlash mumkin.

Prim algoritmining afzalliklari va kamchiliklari

Prim algoritmi graf nazariyasi doirasida ishlatiladigan muhim algoritmlardan biridir. Bu algoritm minimal bog'lanish daraxtini (MST - Minimum Spanning Tree) topish uchun mo'ljallangan. Quyida Prim algoritmining afzalliklari va kamchiliklari haqida ma'lumot beraman:

Afzalliklari:

1. Soddaligi: Prim algoritmi ancha sodda va tushunarli bo'lib, uni amalga oshirish oson.

2. Ishlash tezligi: Kichik grafiklar uchun (masalan, n < 100) Prim algoritmi juda tez ishlaydi. Uning vaqti O(E log V) bo'lishi mumkin, bu yerda E - chekkalar soni, V - tugunlar soni.

3. To‘g‘ridan-to‘g‘ri qo‘llash: Agar grafikni matritsa ko‘rinishida ifodalangan bo‘lsa, bu algoritm juda samarali ishlaydi.

4. Minimal bog'lanish: Har qanday bog'lanish grafikasi uchun minimal bog'lanish daraxtini topishga qodir.

5. Qo'shimcha ma'lumotlarni hisoblash: Algoritmni qo'llash davomida boshqa ko'plab ma'lumotlarni (masalan, eng qisqa yo'l) hisoblash imkoniyatiga ega bo'lishingiz mumkin.

Kamchiliklari:

1. Grafning harakatlanuvchiligi: Agar grafik o'zgaruvchan bo'lsa (masalan, chekkalar vaznlarini o'zgartirish), Prim algoritmini har safar qayta ishlash kerak bo'ladi.

2. Katta grafiklardagi samarasizlik: Katta grafiklar bilan ishlashda (n > 1000) Prim algoritmining samaradorligi pasayishi mumkin.

3. O'zaro aloqa yo'q: Algoritm barcha tugunlar orasidagi eng yaxshi aloqa yo'llarini aniqlamaydi; faqatgina minimal bog'lanishni ta'minlaydi.

4. Yana bir xil natija olish ehtimoli yuqori: Bir xil graf bilan bir necha bor bajarilganda natijalar bir xilda chiqishi mumkin va bu ba’zan noqulaylik tug'diradi.

5. Qo'shimcha joy talab etilishi: Agar priority queue yoki boshqa ma’lumotlar tuzilmalari qo'llanilsa, ular qo'shimcha joy talab qiladi.

Umuman olganda, Prim algoritmi minimal bog'lanish daraxtini topishda samarali va qulay vosita hisoblanadi, lekin uning kamchiliklarini ham inobatga olish zarur.

Zamonaviy texnologiyalar kontekstida Prim algoritmining rivojlanishi

Prim algoritmi, 1930-yillarda Eugene Charles Catalan tomonidan ishlab chiqilgan bo'lib, minimal ostov daraxtini (MST) topish uchun ishlatiladi. Zamonaviy texnologiyalar kontekstida Prim algoritmining rivojlanishi va qo'llanilishi bir qator yo'nalishlarda amalga oshirilmoqda:

1. Ma'lumotlar tuzilishi va algoritmlar

• Samaradorlikni oshirish: Zamonaviy kompyuterlarning quvvati va xotira imkoniyatlari yordamida Prim algoritmining samaradorligi oshirildi. Masalan, prioritet navbatlarini (min-heap) ishlatish orqali algoritmning vaqt murakkabligi O(E log V) ga tushirildi.

• Parallel hisoblash: Zamonaviy ko'p yadroli protsessorlar va GPUlar yordamida Prim algoritmini parallel ravishda amalga oshirish mumkin, bu esa katta grafikalarni tezda hisoblash imkonini beradi.

2. Grafikalar va tarmoqlar

• Katta grafikalar: Internet tarmoqlari, ijtimoiy tarmoqlar va boshqa katta grafikalar ustida ishlashda Prim algoritmi foydalidir. Masalan, ijtimoiy tarmoqdagi eng qisqa yo'llarni aniqlashda yoki ma'lumotlar markazlari o'rtasidagi optimal aloqa yo'llarini topishda qo'llaniladi.

• Transport va logistika: Transport tarmoqlarida eng samarali yo'nalishlarni belgilash uchun Prim algoritmi yordamida minimal xarajatli bog'lanishlarni topish mumkin.

3. Sun'iy intellekt va mashinani o'qitish

• O'qituvchi tizimlar: Sun'iy intellekt va mashinani o'qitish sohalarida Prim algoritmi yordamida ma'lumotlar strukturalarini optimallashtirish va tahlil qilishda qo'llaniladi. Masalan, ma'lumotlar to'plamlaridagi eng muhim xususiyatlarni tanlashda.

• Rekomendatsiya tizimlari: Rekomendatsiya tizimlarida foydalanuvchilar o'rtasidagi aloqalarni grafiklar sifatida ko'rib chiqib, Prim algoritmi yordamida eng samarali tavsiyalarni berish mumkin.

4. Kiberxavfsizlik

• Tarmoq xavfsizligi: Kiberxavfsizlik sohasida tarmoqning minimal bog'lanishlarini aniqlash va xavfsizlikni ta'minlashda Prim algoritmi qo'llanilishi mumkin. Bu, masalan, tarmoqdagi zaif joylarni aniqlashda yordam beradi.

```
using System;
using System.Collections.Generic;
class Program
{    static void Main(string[] args)    {
      // Grafning og'irliklarini belgilash
      int[,] graph = new int[,]       {
         { 0, 2, 0, 6, 0 },
         { 2, 0, 3, 8, 5 },
         { 0, 3, 0, 0, 7 },
         { 6, 8, 0, 0, 9 },
         { 0, 5, 7, 9, 0 }       };
      PrimMST(graph);    }
   static void PrimMST(int[,] graph)
   {       int verticesCount = graph.GetLength(0);
      bool[] inMST = new bool[verticesCount];
      int[] key = new int[verticesCount];
      int[] parent = new int[verticesCount];
      // Barcha kalitlarni cheksizga o'rnatamiz
      for (int i = 0; i < verticesCount; i++)
      {          key[i] = int.MaxValue;          inMST[i] = false;       }
      // Birinchi tugunni tanlaymiz
      key[0] = 0; // Birinchi tugun uchun kalit
      parent[0] = -1; // Birinchi tugun uchun ota tugun yo'q
```

```
        for (int count = 0; count < verticesCount - 1; count++)
        {           // Minimal kalitni tanlash
            int u = MinKey(key, inMST);
            inMST[u] = true;
            // Qo'shni tugunlar orqali o'tish
            for (int v = 0; v < verticesCount; v++)
            {               if (graph[u, v] != 0 && !inMST[v] && graph[u, v] <
ey[v])
                {               parent[v] = u;
                    key[v] = graph[u, v];           }       }       }
        PrintMST(parent, graph);
    }   static int MinKey(int[] key, bool[] inMST)
    {       int min = int.MaxValue;
        int minIndex = -1;
        for (int v = 0; v < key.Length; v++)
        {           if (!inMST[v] && key[v] < min)          {
                min = key[v];
                minIndex = v;           }       }       return minIndex;    }
    static void PrintMST(int[] parent, int[,] graph)
    {       Console.WriteLine("Minimal Ostov Daraxti:");
        for (int i = 1; i < parent.Length; i++)
        {       Console.WriteLine($"Tugun {parent[i]} - Tugun {i}: Og'irlik
{graph[i, parent[i]]}");
        }   }}
```

Kodni tushuntirish:

1. Grafni belgilash: graph massivi orqali grafning og'irliklarini belgilaymiz. 0 qiymati bog'lanmagan tugunlar orasidagi aloqani ifodalaydi.

2. PrimMST metodi: Bu metod minimal ostov daraxtini hisoblash uchun asosiy algoritmni amalga oshiradi.

3. MinKey metodi: Bu metod minimal kalit qiymatini va unga tegishli tugunni topadi.

FOYDALANILGAN ADABIYOTLAR

1. Kleinberg, J.E. Tardos, (2005). "Algorithm Design." Pearson Education.

2. Cormen, T. H. Leiserson, C. E. Rivest, R. L. Stein, C. (2009). "Introduction to Algorithms" (3rd ed.). MIT Press.

3. Sedgewick, R. Wayne, K. (2011). "Algorithms" (4th ed.). Addison-Wesley.

4. Tarjan, R. E. (1983). "Data Structures and Network Algorithms." SIAM.

5. Bertsekas, D. P., Tsitsiklis, J. N. (2000). "Introduction to Network Optimization." Athena Scientific.

7. Kumar, P., Sharma, A. (2019). "A Survey on Minimum Spanning Tree Algorithms." International Journal of Computer Applications.

8. Zhang, Y., Wang, L. (2020). "A Review of Minimum Spanning Tree Algorithms in Big Data Environment." IEEE Access.

9. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. B CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (T. 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.