

TEZKOR INTERNET ORQALI SAMARALI MA'LUMOT UZATISHDA
DEYKSTRI ALGORITMINING ROLI

Farmonov Sherzodbek Raxmatjonovich

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

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

Boqijonova Shaxnoza Zoirjon qizi

Farg'ona davlat universiteti talabasi

[*boqijonovashaxnoza3@gmail.com*](mailto:boqijonovashaxnoza3@gmail.com)

***Anotatsiya:** Maqolada Deykstri algoritmining tezkor internet tarmog'ida ma'lumot uzatish yo'nalishlarini optimallashtirishdagi ahamiyati ko'rib chiqiladi. Algoritmning o'tkazuvchanlik, kechikish va samaradorlikka ta'siri tahlil qilinib, uning tarmoq samaradorligini oshirishdagi roli amaliy misollar bilan yoritiladi.*

***Kalit so'zlar:** Deykstri algoritmi, tezkor internet, ma'lumot uzatish, graflar, yo'nalishni optimallashtirish, o'tkazuvchanlik, kechikish, tarmoq samaradorligi, algoritmning amaliy qo'llanilishi.*

***Annotation:** The article examines the significance of the Dijkstra algorithm in optimizing data transmission paths in high-speed internet networks. It analyzes the algorithm's impact on bandwidth, latency, and efficiency, highlighting its role in enhancing network performance with practical examples.*

***Keywords:** Dijkstra algorithm, high-speed internet, data transmission, graphs, path optimization, bandwidth, latency, network efficiency, practical application of algorithms.*

***Аннотация:** В статье рассматривается значение алгоритма Дейкстры для оптимизации путей передачи данных в высокоскоростных интернет-сетях. Анализируется влияние алгоритма на пропускную*

способность, задержку и эффективность, подчеркивается его роль в повышении производительности сети на практических примерах.

Kirish : Deyksrti algoritmi haqida qisqacha ma'lumot: Deyksrti algoritmi - graflar nazariyasida ishlatiladigan, bir tugundan boshqa barcha tugunlarga bo'lgan eng qisqa yo'lni topish uchun ishlab chiqilgan matematik usul. Bu algoritmni 1956-yilda niderlandiyalik matematik va informatika olimi Edsger Dijkstra kashf qilgan. Uning asosiy maqsadi – yo'lning eng qisqa va samarali variantini topish orqali masofani, vaqtni va xarajatni kamaytirishdir.

Deyksrti algoritmining asosiy printsiplari: Deyksrti algoritmi graflar ustida ishlashda yo'llarning uzunligini tahlil qilish orqali eng qisqa yo'lni aniqlaydi. Bunda grafikdagi har bir yo'lning vazni (weight) asosiy rol o'ynaydi. Vazn - qirra (edge) orqali ikki tugunni (vertex) bog'lovchi yo'lning qiymatini bildiradi. Bu qiymat turli kontekstlarda har xil ma'noni anglatishi mumkin. Misol uchun:

Masofa => Geografik joylar orasidagi kilometrlar.

Vaqt => Maqsadga yetib borish uchun zarur bo'lgan vaqt.

Xarajat => Yo'lga ketadigan moliyaviy xarajat yoki boshqa resurslar.

Algoritm eng qisqa yo'lni aniqlash uchun quyidagilarni hisobga oladi:

1. Boshlang'ich tugun: Algoritm belgilangan boshlang'ich tugundan (source node) boshlanadi. Bu tugundan barcha qo'shni tugunlarga bo'lgan masofa hisoblab chiqiladi.

2. Vaznlarni solishtirish: Har bir qirraga mos bo'lgan vaznlar solishtiriladi, va algoritm eng kam vaznga ega yo'lni tanlaydi.

3. Qo'shni tugunlarni yangilash: Har safar eng qisqa masofaga ega tugun tanlangandan so'ng, uning qo'shni tugunlarining masofalari yangilanadi.

Masalan, agar $A \rightarrow B$ yo'lining masofasi 3 va $A \rightarrow C \rightarrow B$ yo'lining masofasi $2+1=3$ bo'lsa, algoritm ikkala yo'lni teng deb hisoblaydi.

4. Eng qisqa masofani aniqlash: Algoritm tugunlarni birma-bir o'rganib chiqib, boshlang'ich tugundan maqsad tugunga bo'lgan eng qisqa yo'lni aniqlaydi.

Muhim jihatlar:

Vaznlar ta'siri: Algoritm yo'llarning vazniga asoslangan holda ishlaydi, shuning uchun ijobiy vaznlar bilan ishlash muhimdir. Salbiy vaznlar bo'lsa, algoritm noto'g'ri natijalar berishi mumkin. Oldin ko'rilgan tugunlar qayta ishlanmaydi: Bu jarayon hisoblash samaradorligini oshiradi.

Amalda ishlashi: Misol uchun, siz shaharda eng qisqa masofada joylashgan do'konga borishni istaysiz. Algoritm sizning uyingizdan (boshlang'ich tugun) do'konga (maqsad tugun) boradigan barcha yo'llarni tahlil qilib, masofa bo'yicha eng qisqa yo'lni tanlaydi. Bu yo'lda faqat qo'shni tugunlar bilan ishlash orqali hisoblash jarayonini tezlashtiradi. Shunday qilib, Deykstri algoritmi graflardagi eng qisqa yo'lni topishda eng samarali usullardan biri hisoblanadi.

Tezkor internet tarmog'ida Deykstri algoritmining qo'llanishi: Tezkor internet tarmog'ida samarali va tezkor ma'lumot uzatish tarmoq resurslarini optimallashtirishning asosiy shartlaridan biridir. Deykstri algoritmi aynan shu jarayonda, ya'ni ma'lumot paketlarini manzilga eng qisqa va samarali yo'l orqali yetkazishda muhim rol o'ynaydi. Internet tarmog'i graflar nazariyasi asosida modellashtirilishi mumkin, bu esa Deykstri algoritmini qo'llashni juda qulay qiladi.

Internet tarmog'ini graflar sifatida modellashtirish:

- Tugunlar (vertices): Internet tarmog'ida tugunlar serverlar, marshrutizatorlar yoki foydalanuvchi qurilmalari sifatida ifodalanadi.

- Qirralar (edges): Tugunlar orasidagi aloqalar, ya'ni internet kabeli, optik tolalar yoki simsiz ulanishlar qirra vazifasini bajaradi.

- Vaznlar (weights): Bu qirralarning qiymati bo'lib, u turli omillarga qarab belgilanadi:

=> Aloqa yo'lining uzunligi yoki fizik masofa.

=> Kechikish (latency), ya'ni signalning manzildan manzilga yetib borish vaqti.

=> O'tkazuvchanlik (bandwidth), ya'ni bir vaqtning o'zida qancha ma'lumot uzatilishi mumkinligi.

Deykstri algoritmi bu omillarni hisobga olib, tugunlar o‘rtasida ma'lumot uzatish uchun eng samarali yo‘lni tanlaydi.

Deykstri algoritmining tarmoqda ishlash tartibi:

- Boshlang‘ich tugundan boshlash: Ma'lumot paketi yuboriladigan joydan barcha mumkin bo‘lgan yo‘llar tahlil qilinadi.
- Qo‘shni tugunlarni baholash: Aloqa yo‘llarining vaznlari solishtiriladi va eng qisqa vaqt yoki eng kam kechikish talab qiluvchi yo‘l tanlanadi.
- Masofalarni yangilash: Har safar eng qisqa yo‘l topilganida, boshqa yo‘llar ham qayta hisoblanadi. Bu tarmoqdagi trafikni samarali boshqarishga yordam beradi.
- Maqsad tugunga yetib borish: Algoritm barcha tugunlarni ko‘rib chiqqach, ma'lumot paketi eng qisqa yo‘l orqali maqsadga yetadi.

Amaliy misol:

Agar foydalanuvchi bir veb-saytga kirishga urinayotgan bo‘lsa, uning so‘rovi A marshrutizatoridan B serverga yetishi kerak. Deykstri algoritmi quyidagicha ishlaydi:

- 1 – qadam: A va B orasidagi barcha mavjud yo‘llarni tahlil qiladi.
 - 2 – qadam: Har bir yo‘lning kechikish vaqtini, trafik zichligini va o‘tkazuvchanligini hisobga oladi.
 - 3 – qadam: Eng samarali yo‘lni tanlab, so‘rovni shu yo‘l orqali yuboradi.
- Masalan, $A \rightarrow C \rightarrow B$ yo‘li 20 millisekund bo‘lsa, $A \rightarrow D \rightarrow B$ yo‘li esa 30 millisekundni tashkil etadi. Dijkstra algoritmi 20 millisekundlik yo‘lni tanlaydi.

Internet tarmog‘ida algoritmning afzalliklari:

- Ma'lumot uzatish tezligi oshadi: Ma'lumot paketlari eng qisqa yo‘l orqali yetkaziladi.
- Trafik boshqaruvi: Tarmoqda yuklanish muammolari kamayadi, chunki ortiqcha vaqt talab qiluvchi yo‘llar avtomatik ravishda chetlab o‘tiladi.
- Resurslarni tejash: Tarmoq resurslaridan maksimal darajada samarali foydalanish imkoniyati yaratiladi.

Real hayotda algoritmning qo'llanilish misollari: Deykstri algoritmi faqat nazariy jihatdan emas, balki ko'plab texnologik va kundalik hayotiy sohalarida muvaffaqiyatli qo'llaniladi. Quyida uning tezkor internet tarmog'idagi va boshqa sohalaridagi amaliy qo'llanilishiga misollar keltiriladi:

1. Internet provayderlar va ma'lumot uzatish tarmoqlari

Internet provayderlari Deykstri algoritmidan tarmoqlar orasidagi trafikni samarali boshqarishda foydalanadi. Masalan:

⇒ Internet so'rovlarida: Agar foydalanuvchi biror saytga kirishni xohlasa, provayderning tizimi ma'lumot paketlarini foydalanuvchidan maqsadli serverga eng qisqa va tezkor yo'l orqali uzatadi.

⇒ Kechikishni kamaytirishda: Agar tarmoqda ortiqcha yuklanish (trafik) bo'lsa, algoritm boshqa kamroq yuklangan yo'llarni tanlash orqali ma'lumotni uzatadi.

2. Global CDN (Content Delivery Network) tizimlari

CDNlar foydalanuvchilarga veb-saytlar va videolarni tezkor yetkazish uchun tarmoqdagi eng qisqa yo'llarni tanlaydi. Masalan:

⇒ Netflix yoki YouTube kabi xizmatlar foydalanuvchining joylashuviga qarab eng yaqin serverni topish uchun Deykstri algoritmidan foydalanadi. Bu ma'lumot yuklanish vaqtini qisqartiradi va videolarni uzluksiz tomosha qilishni ta'minlaydi.

3. Transport va logistika tizimlari

Internetga bog'liq transport ilovalari, masalan, Google Maps, Waze yoki Yandex.Navigator, marshrutlarni rejalashtirishda Dijkstra algoritmini ishlatadi. Misol uchun:

⇒ Tugunlar sifatida shaharlar yoki chorrahalar, qirralar sifatida esa yo'llarni qabul qilib, eng qisqa masofani topadi.

⇒ Kechikish vaqtini hisobga olib, tirbandliklarni aylanib o'tadigan yo'llarni tavsiya qiladi.

4. IoT (Internet of Things) qurilmalari

Zamonaviy aqlli qurilmalar tarmoq ichidagi resurslarni samarali taqsimlash uchun Dijkstra algoritmiga tayanadi. Masalan:

⇒ Aqlli uy tizimlarida ma'lumotlar o'zaro ulanishlarda eng tezkor yo'llar orqali uzatiladi.

⇒ Sensor tarmoqlari orasidagi kechikish vaqtini kamaytiradi va o'tkazuvchanlikni oshiradi.

Deykstri algoritmining cheklovlari:

Har qancha foydali bo'lmasin, Deykstri algoritmi ma'lum bir cheklovlarga ega:

1. Salbiy vaznli qirralar bilan ishlamasligi: Algoritm faqat ijobiy vaznlarga ega graflarda ishlaydi. Salbiy vaznlar bilan ishlash uchun Bellman-Ford algoritmi talab etiladi.

2. Katta graflarda resurs iste'moli: Juda katta graflar uchun algoritmni ishlatish vaqt va xotira jihatidan samarasiz bo'lishi mumkin.

3. Tarmoq o'zgaruvchanligi: Real vaqtda tarmoqda trafik sharoitlari tez-tez o'zgarishi mumkin, bu esa hisob-kitobni qayta qilishni talab qiladi.

Masala: Tarmoqda eng qisqa yo'l orqali xabar yuborish

Xonada bir nechta kompyuterlar(tugunlar) mavjud va ular bir – biriga ulanib turadi. Kompyuter A dan E ga xabar yuborish uchun eng qisqa yo'lni topish talab qilinadi.

Tarmoq diagrammasi:

⇒ Kompyuterlar: A, B, C, , D, E

⇒ Aloqalar (vaznlar): ◦ A → B : 3 ms

◦ A → C : 1 ms

◦ B → D : 4 ms

◦ C → D : 2ms

◦ D → E : 1 ms

Deykstri algoritmi C# kodida:

`using System;`

```

internal class Program
{
    public static void Main(string[] args) { // Tarmoq grafigi
        var graph = new Dictionary<string, Dictionary<string, int>>
        {
            { "A", new Dictionary<string, int> { { "B", 3 }, { "C", 1 } } },
            { "B", new Dictionary<string, int> { { "D", 4 } } },
            { "C", new Dictionary<string, int> { { "D", 2 } } },
            { "D", new Dictionary<string, int> { { "E", 1 } } },
            { "E", new Dictionary<string, int>() };
        }
        // Boshlang'ich va maqsad tugunlar
        string start = "A";
        string target = "E";
        var result = Dijkstra(graph, start, target);
        Console.WriteLine($"Eng qisqa yo'l: {result.path}");
        Console.WriteLine($"Umumiy kechikish: {result.distance}ms");
        // Deykstri algoritmi
        static (string path, int distance) Dijkstra(Dictionary<string,
Dictionary<string, int>> graph, string start, string target) {
            var distances = new Dictionary<string, int>();
            var previousNodes = new Dictionary<string, string>();
            var unvisited = new List<string>();
            // Boshlang'ich sozlash
            foreach (var node in graph.Keys)
            {
                distances[node] = int.MaxValue;
                previousNodes[node] = null;
                unvisited.Add(node);
            }
            distances[start] = 0; while (unvisited.Count > 0) {
                // Eng yaqin tugunni topish
                string currentNode = null;
                foreach (var node in unvisited) {
                    if (currentNode == null || distances[node] < distances[currentNode])

```

```
{currentNode = node;}}
    unvisited.Remove(currentNode);
    // Qo'shni tugunlarni yangilash
    foreach (var neighbor in graph[currentNode]){
    int newDist = distances[currentNode] + neighbor.Value;
    if (newDist < distances[neighbor.Key]){
    distances[neighbor.Key] = newDist;
    previousNodes[neighbor.Key] = currentNode;}}}}
    // Yo'lni tiklash
    var path = new Stack<string>();
    var current = target;
    while (current != null){
    path.Push(current);
    current = previousNodes[current];}
    return (string.Join(" -> ", path), distances[target]);}}}}
```

Natija: Eng qisqa yo'l: A -> C -> D -> E

Umumiy kechikish: 4ms

Maqolada Dijkstra algoritmining tezkor internet tarmoqlarida ma'lumot uzatishni optimallashtirishdagi ahamiyati tahlil qilindi. Algoritm grafikdagi eng qisqa yo'lni topish orqali ma'lumotlarni minimal kechikish va maksimal samaradorlik bilan uzatishga yordam beradi.

Tahlillar ushbu algoritmnining tarmoq samaradorligini oshirishdagi muhim rolini ko'rsatdi. Shunday qilib, Dijkstra algoritmi murakkab tizimlarni optimallashtirishda samarali yechimlar yaratishda asosiy vosita hisoblanadi.

Foydalanilgan adabiyotlar ro'yhati:

1. San José Vissiers, I. (2018). Algoritmo de Dykstra.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press. ISBN: 978-0262046305.
3. Sedgewick, R., & Wayne, K. (2011). Algorithms. Addison-Wesley. ISBN: 978-0132762564.

4. P. Хаггарти, «Дискретная математика для программистов» [2003, RUS]
5. Skiena, S. S. (2008). The Algorithm Design Manual. Springer. ISBN: 978-1848000698.
6. Kumar, D. (2018). Graph Algorithms in Python. Packt Publishing. ISBN: 978-1786464858.
7. Joseph Albahari (2023) C# 12 in a Nutshell: The Definitive Reference 1st Edition, Kindle. O'Reilly 2023
8. García, R. C. (2018). Contributions to the theory and applications of projection algorithms (Doctoral dissertation, Universidad de Murcia).
9. Kabilovna, M. D., Muhammadovich, M. F., & Marifjonovich, S. D. (2023). JAMOAT TRANSPORTI MARSHRUTLARINI QURISH INTELLEKTUAL ALGORITMLARI. Al-Farg'oniy avlodlari, 1(4), 95-103.
10. Kabilovna, M. D., Muhammadovich, M. F., & Marifjonovich, S. D. (2023). JAMOAT TRANSPORTI MARSHRUTLARINI QURISH INTELLEKTUAL ALGORITMLARI. Al-Farg'oniy avlodlari, 1(4), 95-103.
11. Yusupova, J. K. (2023). BOG 'LANGAN GRAFLARDA MARSHRUTLAR. ENG QISQA MARSHRUTNI ANIQLASH ALGORITMI. UNI VARIANTLAR SONI BO 'YICHA XAJMINI BAXOLASH. Educational Research in Universal Sciences, 2(2), 258-262.
12. Karimov, N. (2023). SHAHAR TRANSPORTLARI OPTIMAL YO'NALISHNI TANLASHNING AXBOROT TIZIMINING MOBIL NAQLINI YARATISHDA DEYKSTRA VA A STAR ALGORITMLARI TAHLILI. Scientific progress, 4(2), 271-280.
13. Farmonov, S., & Rustamova, N. (2024, May). SINFLASHNING METRIK ALGORITMLARI, YAQIN QO'SHNI USULI VA UNI UMUMLASHTIRISH HAMDA ULARNI NEYRON TARMOQ TEXNOLOGIYALARIDA QO'LLANILISHI. In Международная конференция академических наук (Vol. 3, No. 5, pp. 71-75).
14. Farmonov, S., & Ergashaliyeva, B. (2024). QATIYMAS NEYRON TO'RLAR: MAMDANI QATIYMAS MANTIQUIY XULOSASI, SUGENO

QAT'IYMAS MANTIQIY XULOSASI. Development and innovations in science, 3(5), 62-70.

15. Farmonov, S. (2023). C# Dasturlash Tilida Gray Kodi Bilan Ishlash. Центральноазиатский журнал образования и инноваций, 2(12 Part 2), 71-74.

16. Farmonov, S., & Rasuljonova, Z. (2024). Ob'Ektga Yo'Naltirilgan Dasturlash Zamonaviy Dasturlashning Asosi Sifatida. Центральноазиатский журнал образования и инноваций, 3(1), 83-86.

17. Farmonov, S., & Usmonaliyev, U. (2024). O'ZBEKISTON RESPUBLIKASI IT SOHASINING RIVOJLANISH ISTIQBOLLARI. Бюллетень педагогов нового Узбекистана, 2(1), 59-62.

18. Raxmonjonovich, F. S. (2023). Dasturlashda Istisnolarning Ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 475-481.

19. GeeksforGeeks. (n.d.). *Dijkstra's Shortest Path Algorithm*. Retrieved from <https://www.geeksforgeeks.org>

20. Stanford University. (n.d.). *Algorithms Specialization*. Retrieved from Coursera: <https://www.coursera.org>.