

ENG YAQIN SHAHARLARNI ANIQLASHDA GEOMETRIK ALGORITMLARNING AHAMIYATI

Farmonov Sherzodbek Rahmonjonovich

Fargʻona davlat universiteti

amaliy matematika va informatika kafedrası katta oʻqituvchisi

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

Muhammadsodiqova Guljahon Azizjon qizi

Fargʻona davlat universiteti 2-kurs talabasi

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

Annotatsiya: *Ushbu maqolada geometrik algoritmlarning "eng yaqin shaharni topish" masalasidagi qoʻllanilishi koʻrib chiqilgan. Masala yechimida Voronoy diagrammalari va Evklid masofasidan foydalanilgan. Algoritm har bir shahar uchun markaziy nuqta bilan masofani hisoblaydi va eng yaqin shaharni aniqlaydi.*

Kalit soʻzlar: *Geometrik algoritmlar, Voronoy diagrammasi, eng yaqin shahar, Evklid masofasi, masofani hisoblash.*

Аннотация: *В данной статье рассмотрено применение геометрических алгоритмов к задаче «нахождения ближайшего города». Для решения использованы диаграммы Вороного и евклидово расстояние. Алгоритм вычисляет расстояние от каждой точки до центра города, определяет ближайший.*

Ключевые слова: *Геометрические алгоритмы, диаграмма Вороного, ближайший город, евклидово расстояние, расчет расстояния.*

Annotation: *This article explores the application of geometric algorithms to the "nearest city determination" problem. The solution employs Voronoi diagrams and Euclidean distance. The algorithm calculates the distance from each point to the city center and identifies the nearest one.*

Keywords: *Geometric algorithms, Voronoi diagram, nearest city, Euclidean distance, distance calculation.*

Geometrik algoritmlar — bu matematik va kompyuter fanlarida o'rganiladigan algoritmlar bo'lib, ular geometrik shakllar, fazoviy obyektlar va ularning o'zaro joylashuvi bilan bog'liq masalalarni hal qiladi. Bu algoritmlar asosan tekislikda yoki fazoda geometrik obyektlar bilan ishlash masalan, nuqtalar, chiziqlar, poligonlar, sferalar kabi obyektlar o'rtasidagi masofalar aniqlashda qo'llaniladi. Geometrik algoritmlarning maqsadi fazoviy obyektlar orasidagi o'zaro bog'lanishlarni aniq va samarali tarzda hisoblashdir. Bular ko'plab amaliy sohalarda masalan, kompyuter grafikasida, robototexnikada, geografik axborot tizimlarida (GIS), kompyuter tizimlarida joylashuv aniqlashda qo'llaniladi.

Geometrik algoritmlar asosida turgan asosiy masalalar:

1. Kesishish masalalari: Geometrik obyektlar, masalan, chiziqlar, poligonlar yoki doiralar orasidagi kesishishlarni aniqlash.
2. Tartiblash va to'plamlar: Geometrik obyektlar to'plamini o'zaro joylashuvi va ularning tartibiga asoslanib saralash.
3. Qavslar (Convex Hull): Berilgan nuqtalar to'plamining qavsini (eng kichik poligon) topish.
4. Voronoy diagrammalari: Nuqtalar to'plamining Voronoy diagrammasini yaratish, ya'ni har bir nuqtaning eng yaqin hududini aniqlash.

Voronoy Diagrammasi algoritmining ishlash jarayoni. Voronoy diagrammasi algoritmi bir nechta nuqtalar (masalan, shaharlar) berilgan holda, har bir nuqtaga eng yaqin hududni ajratish uchun ishlatiladi. Ushbu diagramma har bir nuqtani o'ziga yaqin bo'lgan hududga ajratadi. Voronoy diagrammasi yordamida har bir nuqta o'zining "bo'linmasi" ga ega bo'ladi, ya'ni, uning atrofida hudud faqat shu nuqtaga tegishli bo'ladi.

Algoritmning ishlash jarayoni:

- 1. Nuqtalarni aniqlash:** Avvalo, Voronoy diagrammasi yaratish uchun bir nechta nuqtalar (masalan, shaharlar, xizmat ko'rsatuvchi markazlar yoki boshqa

ob'ektlar) beriladi. Har bir nuqtaning o'zining koordinatalari ($x ; y$) mavjud bo'ladi.

2. Masofalar hisoblanadi: Har bir nuqta o'ziga eng yaqin bo'lgan hududni aniqlash uchun barcha nuqtalar orasidagi masofalar hisoblanadi. Masofa ikki nuqta o'rtasidagi masofa formula orqali hisoblanadi. Bu masofa har bir nuqtani boshqalardan ajratish uchun kerak bo'ladi.

3. Voronoy Diagrammasining chizilishi: Keyin barcha nuqtalar o'rtasidagi masofalar hisoblangach diagramma chiziladi. Diagrammada har bir nuqtaga tegishli bo'lgan hudud (ya'ni, Voronoy hududi) ajratiladi. Hududni ajratish uchun nuqtalar o'rtasidagi teng masofa chiziqlari chiziladi. Bu chiziqlar nuqtalar orasida joylashgan "**bo'linma**" yoki "**chegara**" bo'ladi.

4. Hududlarni taqsimlash: Voronoy diagrammasida har bir nuqtaning atrofidagi hudud aniq belgilangan bo'ladi. Bunda har bir nuqta faqat o'ziga eng yaqin bo'lgan hududga tegishli bo'ladi. Diagrammada har bir nuqta o'zining "hududiga" ega bo'ladi ya'ni bu nuqtaning atrofidagi hudud faqat shu nuqtaga tegishli bo'ladi.

5. Yaqin nuqtalarni aniqlash: Diagramma chizilganidan so'ng, endi har bir nuqtaning o'ziga eng yaqin nuqtasi aniqlanadi. Masalan, agar biror nuqtadan (shahar, xizmat ko'rsatuvchi markaz va hokazo) eng yaqin nuqtani (masalan, boshqa shaharni) topish uchun Voronoy diagrammasidagi o'z hududi orqali bu nuqtani aniqlash mumkin. Har bir nuqta o'zining Voronoy hududida joylashgan nuqtalar orasidan eng yaqin bo'lgan nuqtani topadi.

Voronoy Diagrammasining matematik tavsifi. Voronoy diagrammasi berilgan nuqtalar to'plamiga asoslanib har bir nuqtaning o'ziga eng yaqin bo'lgan hududini aniqlash uchun ishlatiladigan geometrik tuzilma. Bu diagrammada har bir nuqta o'zining Voronoy hududiga ega bo'ladi ya'ni, har bir nuqtaga tegishli bo'lgan hududni belgilash uchun masofalar asosida bo'linmalar chiziladi.

Voronoy diagrammasining formulasi. Voronoy diagrammasi ($P = \{P_1, P_2, \dots, P_n\}$) deb ataluvchi nuqtalar to'plamiga asoslanadi bunda har bir nuqta (

$P_i = (x_i, y_i)$) ikki o'ldovli fazoda joylashgan. Har bir nuqtaning o'ziga tegishli Voronoy hududi quyidagicha matematik tarzda aniqlanadi.

1. Voronoy hududining matematik tasviri:

Har bir nuqta (P_i) o'zining Voronoy hududiga ega. Bu hudud shu nuqtaga eng yaqin bo'lgan barcha nuqtalar uchun tasvirlanadi. Agar (P_j) nuqtasi (P_i) dan yaqin bo'lsa, u holda (P_j) nuqtasi (P_i) nuqtasining Voronoy hududida joylashgan bo'ladi. Matematik tasviriy shaklida:

$$V(P_i) = \{P_j \mid d(P_i, P_j) \leq d(P_i, P_k)\}$$

Bu yerda: $V(P_i)$ - (P_i) nuqtasining Voronoy hududi.

$d(P_i, P_j)$ - (P_i) va (P_j) nuqtalari orasidagi yevklid masofasi.

$d(P_i, P_k)$ - (P_i) va boshqa nuqtalar orasidagi masofa.

2. Masofaning hisoblanishi: Har bir nuqta o'rtasidagi masofa quyidagi evklid formulasi bilan hisoblanadi:

$$d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Bu formula orqali har bir nuqta o'rtasidagi masofa hisoblanadi. Shuningdek, bu masofa Voronoy diagrammasini qurishda asosiy rol o'ynaydi chunki har bir nuqta o'zining eng yaqin bo'lgan nuqtasiga tegishli hududni belgilaydi.

3. Voronoy diagrammasidagi bo'linmalar: Voronoy diagrammasi fazoni har bir nuqtaga tegishli bo'lgan Voronoy hududlarga ajratadi. Bu bo'linmalar nuqtalar orasidagi masofalar teng bo'lgan joylarda chiziladi. Har ikkala nuqtaning masofasi teng bo'lgan chiziqlar Voronoy diagrammasining chiziqlarini tashkil etadi. Matematik tasviriy shakli:

$$\{x \mid d(P_i, x) = d(P_j, x)\}$$

Bu chiziqlar (P_i) va (P_j) nuqtalari orasidagi teng masofalarni ifodalaydi va diagrammada bo'linma chiziqlari sifatida ko'rsatiladi. Diagrammada bu chiziqlar

Voronoy diagrammasining chegara nuqtalarini belgilaydi.

Voronoy Diagrammasining xususiyatlari. Daraxt strukturalari: Voronoy diagrammasi ko‘plab amaliy masalalarda masalan, daraxt strukturalari (trees) va yaqinlikni topish muammolarida qo‘llaniladi. Diagramma masalan, xaritalarda, shaharlar orasidagi masofalarni hisoblashda juda foydali bo‘ladi.

Qayta ishlash va optimallashtirish: Agar nuqtalar soni juda ko‘p bo‘lsa Voronoy diagrammasini yaratish uchun samarali algoritmlar masalan, **Divide and Conquer** yoki **Sweep Line Algorithm** ishlatiladi. Bu algoritmlar diagramma hosil qilishni optimallashtiradi va samarali hisoblashni ta'minlaydi.

Amaliyotdagi qo‘llanilish sohalari:

1. Qidiruv tizimlari va optimallashtirish. Voronoy diagrammalarini qidiruv tizimlarida qo‘llash mumkin. Masalan, geografik joylashuv asosida eng yaqin ma'lumotlarni topish (masalan, foydalanuvchiga eng yaqin restoran yoki do‘koni ko‘rsatish). Voronoy diagrammasi yordamida har bir nuqta uchun "eng yaqin" joylar aniqlanadi va foydalanuvchining joylashuvi bo‘yicha eng optimal javoblar taqdim etiladi.

2. Robototexnika. Robototexnikada Voronoy diagrammalaridan robotlarning harakatini rejalashtirishda foydalaniladi. Masalan, robotlar bir-biridan masofada joylashgan nuqtalarda harakat qilishi kerak bo‘lganda Voronoy diagrammasi yordamida har bir robotning o‘z hududini aniqlash va boshqalardan to‘g‘ri masofada harakat qilishini ta'minlash mumkin.

3. Telekommunikatsiya tarmoqlari. Voronoy diagrammalarini mobil telefon tarmoqlarida qo‘llash mumkin. Masalan, antennalar yoki simsiz tarmoqlarni joylashtirishda Voronoy diagrammasi yordamida har bir stansiya uchun maksimal qamrov hududi va optimal tarmoq joylashuvi aniqlanadi. Bu tarmoqning samarali ishlashini va signalning maksimal darajada bo‘lishini ta'minlaydi.

4. Geografik axborot tizimlari (GIS). Geografik axborot tizimlarida Voronoy diagrammalaridan foydalanish mumkin. Masalan, mintaqadagi yashash punktlarining joylashuvi yoki tabiiy resurslar (masalan, suv manbalari) ning joylashuvi asosida hududlarni ajratish uchun. Bu masalan, suv ta'minoti

tarmoqlarini rejalashtirishda shuningdek, tabiiy resurslardan samarali foydalanishda qo‘llanadi.

5. Nodavlat tashkilotlar va logistika. Voronoy diagrammasi yordamida saqlash omborlari yoki yetkazib berish punktlari o‘rtasida optimal tarqatish tarmoqlarini yaratish mumkin. Bu shuningdek, tezkor yetkazib berish, samarali tashish va saqlash tizimlarini yaratishda ham qo‘llaniladi.

Masala. Bir mintaqada 9 ta shahar joylashgan: Nurafshon, Guliston, Jizzax, Zarafshon, Qo‘qon, Marg‘ilon, Shahrisabz, Termiz, Qarshi. Har bir shaharda tez yordam stansiyasi joylashgan. Mintaqada yangi tez yordam stansiyasini qurish rejalashtirilmogda. Ushbu stansiya shunday joylashishi kerakki, u boshqa barcha shaharlardan eng yaqin masofada turgan aholi punktiga xizmat ko‘rsatsin.

Berilganlar. Shaharlar koordinatalari: Nurafshon: (200, 800), Guliston: (400, 600), Jizzax: (600, 500), Zarafshon: (1000, 200), Qo‘qon: (300, 900), Marg‘ilon: (350, 950), Shahrisabz: (800, 300), Termiz: (850, 250), Qarshi: (320, 870). Yangi tez yordam stansiyasini qurish uchun Voronoy diagrammasi asosida eng yaqin xizmat hududini aniqlash kerak.

Masalaning C# tilidagi ko‘rinishi:

```
using System; namespace Voronoy_diagrammasi { internal class Progra {
// Shahar koordinatalarini saqlash uchun oddiy sinf class Shahar {
    public string Nomi { get; set; }
    public double X { get; set; }
    public double Y { get; set; }
    public Shahar(string nomi, double x, double y){ Nomi = nomi;X = x;Y
= y; } } // Masofa hisoblovchi funksiya static double Masofa(Shahar a, Shahar b){
    return Math.Sqrt(Math.Pow(a.X - b.X, 2) + Math.Pow(a.Y - b.Y,
2));} static void Main(string[] args){ // Shaharlarning koordinatalari List<Shahar>
shaharlar = new List<Shahar>{ new Shahar("Nurafshon", 200, 800),new
Shahar("Guliston", 400, 600),new Shahar("Jizzax", 600, 500),new
Shahar("Zarafshon", 1000, 200),new Shahar("Qo'qon", 300, 900),new
```

```
Shahar("Marg'ilon", 350, 950), new Shahar("Shahrisabz", 800, 300), new
Shahar("Termiz", 850, 250), new Shahar("Qarshi", 320, 870) };
```

```
// Yangi stansiya joylashuvi uchun markaziy nuqtasini toppish double
jamiX = 0, jamiY = 0; foreach (var shahar in shaharlar)
{ jamiX += shahar.X; jamiY += shahar.Y; } // O'rtacha koordinatalarni hisoblash
double ortachaX = jamiX / shaharlar.Count; double ortachaY = jamiY /
shaharlar.Count; // O'rtacha nuqta bilan eng yaqin shaharni topish
Shahar engYaqinShahar = null; double engKichikMasofa = double.MaxValue;
foreach (var shahar in shaharlar) { double masofa = Masofa(new Shahar("Yangi Stansiya",
ortachaX, ortachaY), shahar); if (masofa < engKichikMasofa) { engKichikMasofa
= masofa; engYaqinShahar = shahar; } } // Natijalarni chiqarish
Console.WriteLine("Yangi stansiya joylashuvi:");
Console.WriteLine($"Koordinatalar: ({ortachaX:F2}, {ortachaY:F2})");
Console.WriteLine($"Eng yaqin shahar: {engYaqinShahar.Nomi}, masofa:
{engKichikMasofa:F2} km"); Console.ReadKey(); }
```

Natija: Yangi stansiya joylashuvi:

Koordinatalar: (535,56, 596,67)

Eng yaqin shahar: Jizzax, masofa: 116,18 km

Maqolada geometrik algoritmlarning amaliy masalalardagi ahamiyati ko'rib chiqildi. "Eng yaqin shaharni aniqlash" misolida Voronoy diagrammalari va Evklid masofasidan foydalanilgan algoritmning ishlash jarayoni va samaradorligi tahlil qilindi. Ushbu yondashuv transport va logistika sohalarida muhim qo'llanmalarga ega bo'lib, kelgusida algoritmlarni yanada rivojlantirish imkoniyatini beradi.

FOYDALANILGAN ADABIYOTLAR RO'YXATI

1. Knuth, D. E. (1973). *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley.
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. Azizov, M., & Saidov, S. (2017). **Geometrik algoritmlar va ularning amaliyotdagi qo‘llanilishi.** Tashkent State University of Economics.
5. Turing, A. M. (1936). **On Computable Numbers, with an Application to the Entscheidungsproblem**. Proceedings of the London Mathematical Society, 2(42), 230-265.
6. Shakhnoza, A., & Abdurahmonov, S. (2020). **Geometric Algorithms and Their Applications**. Science Journal of Applied Mathematics and Statistics, 8(2), 104-111
7. Uysal, S. (2019). *Fraktal geometri ve algoritma ilişkisinde jeneratif sanat* (Master's thesis, Güzel Sanatlar Enstitüsü).
8. Lubis, A. P. (2024). Analisis Kemampuan berpikir komputasi siswa SMA menggunakan software Geogebra materi Transformasi Geometri. Journal of International Multidisciplinary Research, 2(5), 24-31.
9. Raxmonjonovich, F. S. (2024). C# VA MASHINA TILI. Ta'lim innovatsiyasi va integratsiyasi, 12(1), 59-62.
10. Farmonov, S., & Usmonaliyev, U. (2024). O'ZBEKISTON RESPUBLIKASI IT SOHASINING RIVOJLANISH ISTIQBOLLARI. Бюллетень педагогов нового Узбекистана, 2(1), 59-62.
11. Farmonov, S. R., & qizi Xomidova, M. A. (2024). C# VA JAVA DASTURLASH TILLARIDA FAYLLAR BILAN ISHLASHNING TURLI USULLARINING SAMARADORLIGI HAQIDA. Zamonaviy fan va ta'lim yangiliklari xalqaro ilmiy jurnal, 1(9), 45-51.
12. Wikipedia – Geometric algorithms, Voronoi diagram. (https://en.wikipedia.org/wiki/Geometric_algorithms, https://en.wikipedia.org/wiki/Voronoi_diagram)
13. Geeks for Geeks – Algorithms for finding nearest points and Voronoi diagram. (<https://www.geeksforgeeks.org>) Stack Overflow – Algorithm discussions and implementation tips. (<https://stackoverflow.com>)