Київський національний університет імені Тараса Шевченка

**Лабораторна робота**

**З навчального курсу «Обчислювальна геометрія та комп’ютерна графіка»**

**Варіант 7**

Виконав:

студент 3 курсу

факультету кібернетики

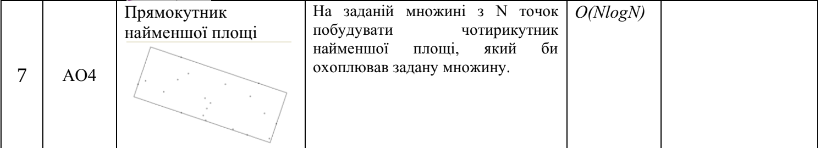
спеціальність «Комп’ютерні науки»

групи ТТП-31

Таран Владислав Віталійович

Київ 2024

**Умова**

****

**Вступ**

Лабораторна робота присвячена розробці алгоритму для обчислення найменшого описуючого прямокутника для заданої опуклої оболонки точок. Найменший описуючий прямокутник є прямокутником, який має мінімальну площу та повністю охоплює опуклу оболонку.

**Алгоритм**

Алгоритм для знаходження найменшого описуючого прямокутника для опуклої оболонки включає наступні кроки:

**Крок 1: Обчислення опуклої оболонки**

Для обчислення опуклої оболонки можна використати алгоритм Грехема зі складністю O(N\*LOG(N))

У цьому розв'язку ми використовуємо функцію compute\_convex\_hull(points), яка обчислює опуклу оболонку за допомогою такого алгоритму.

**Крок 2: Обчислення орієнтації ребра**

Для обчислення орієнтації ребра між двома точками використовується функція compute\_orientation(curr\_point, next\_point).

Вона використовує atan2 для обчислення кута між відрізком, утвореним поточною точкою та наступною точкою, і осі x.

**Крок 3: Обертання точок**

Для обчислення найменшого описуючого прямокутника ми обертаємо опуклу оболонку так, щоб вона була паралельна осі x.

Функція rotate\_points(points, angle) використовується для обертання точок опуклої оболонки навколо початку координат за заданим кутом.

**Крок 4: Обчислення найменшого описуючого прямокутника**

Після обертання опуклої оболонки ми обчислюємо мінімальну та максимальну координату x і y для отриманої оболонки.

Площа найменшого описуючого прямокутника обчислюється як різниця між максимальною та мінімальною координатою x, помножена на різницю між максимальною та мінімальною координатою y.

**Функції**

У реалізації алгоритму використовуються наступні функції:

compute\_convex\_hull(points): Функція для обчислення опуклої оболонки

compute\_orientation(curr\_point, next\_point): Функція для обчислення орієнтації ребра між двома точками за допомогою atan2.

rotate\_points(points, angle): Функція для обертання точок опуклої оболонки навколо початку координат за заданим кутом.

compute\_smallest\_surrounding\_rectangle(points): Основна функція, яка обчислює найменший описуючий прямокутник для заданої опуклої оболонки.

Решта функцій – допоміжні для UI, побудови графіків OnClick подій

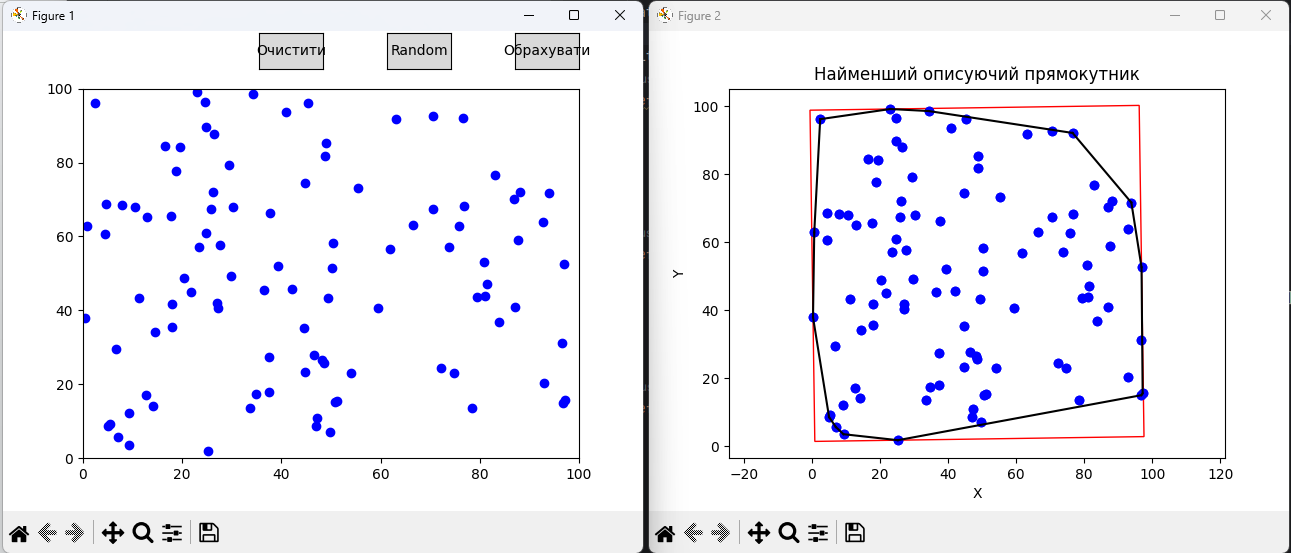
**Складність**

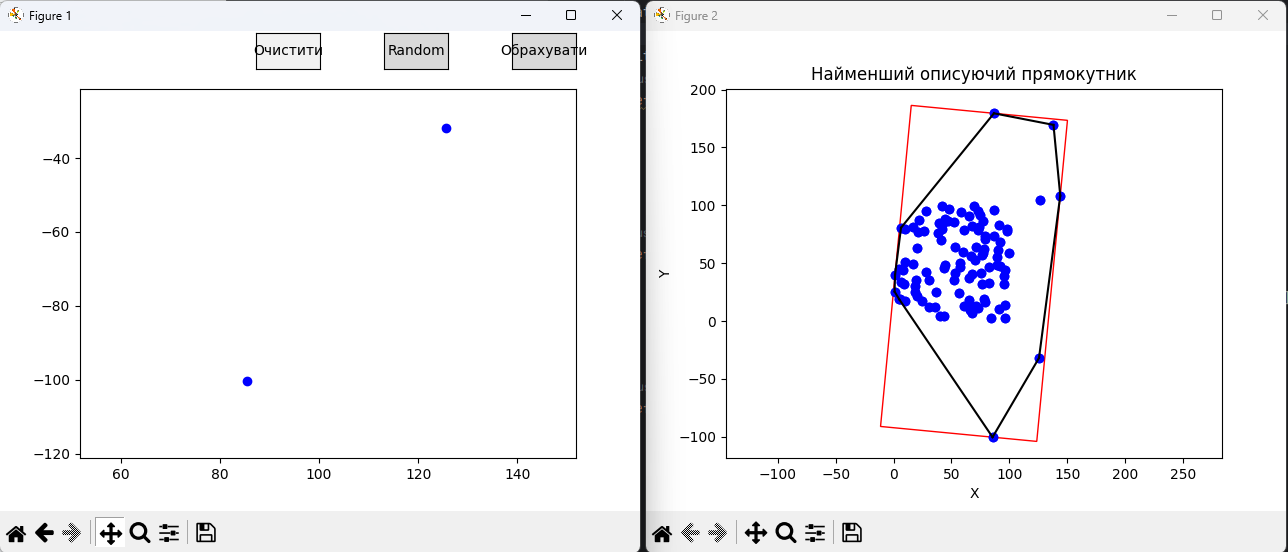
Складність алгоритму для обчислення найменшого описуючого прямокутника залежить від використаного алгоритму для обчислення опуклої оболонки. Якщо для обчислення опуклої оболонки використовується ефективний алгоритм, такий як "QuickHull" або Грехема, то складність обчислення опуклої оболонки становить O(N\*LOG(N)), де N - кількість точок у вхідному наборі.

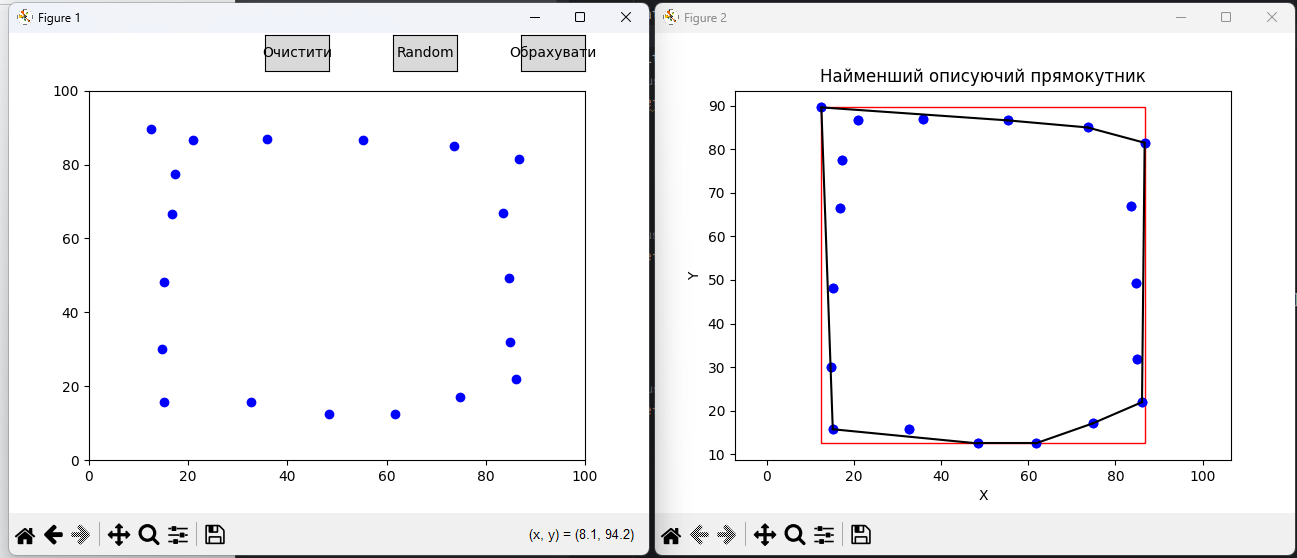
Після отримання опуклої оболонки, обчислення найменшого описуючого прямокутника вимагає ітерації по ребрам опуклої оболонки та обчислення мінімальної площі для кожного ребра. Оскільки кількість ребер опуклої оболонки обмежена, складність цього кроку є константною, тобто O(N).

Таким чином, загальна складність алгоритму для обчислення найменшого описуючого прямокутника залежить від складності обчислення опуклої оболонки, що становить O(N\*LOG(N)), де N - кількість точок у вхідному наборі.

**Приклади роботи програми:**







**Висновки**

У цій лабораторній роботі ми розробили алгоритм для обчислення найменшого описуючого прямокутника для заданої множини точок. Алгоритм включає обчислення опуклої оболонки, обчислення орієнтації ребра, обертання точок та обчислення найменшого описуючого прямокутника. Складність алгоритму залежить від використаного алгоритму для обчислення опуклої оболонки, але загальна складність становить O(N\*LOG(N)), де N - кількість точок у вхідному наборі. Результати показують, що розроблений алгоритм ефективний і може бути успішно використаний для обчислення найменшого описуючого прямокутника для опуклої оболонки точок у практичних завданнях.