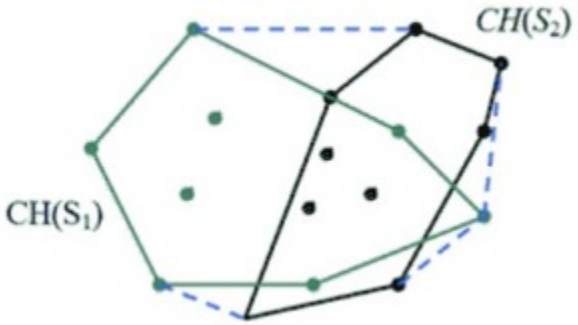
Грищенко Юрій ІПС-32
Лабораторна робота №6

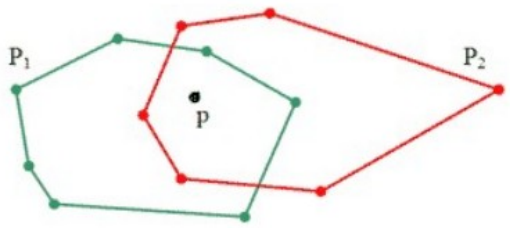
Постановка задачі: Знайти опуклу оболонку для множини точок X — мінімальну опуклу множину, що містить X.

Опис алгоритму:

Алгоритм типу “розділяй та владарюй” (Шеймос) — рекурсивний алгоритм, дозволяє знайти опуклу оболонку для двох рівних підмножин X і швидко об’єднати їх в одну оболонку.

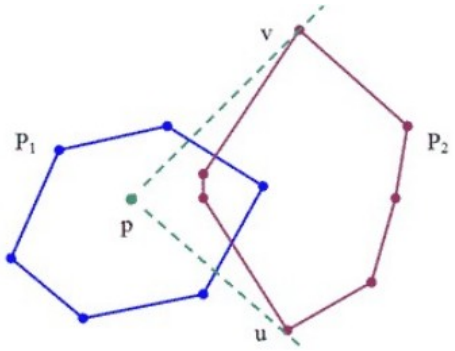


1. Якщо кількість точок множини маленька, знаходимо оболонку якимось простішим методом (наприклад, Джарвіса) і повертаємо її.
2. Інакше, розділяємо множину на рівні підмножини S_1, S_2 і рекурсивно для них викликаємо алгоритм Шеймоса, знаходимо опуклі оболонки P_1, P_2
3. Знаходимо деяку внутрішню точку S_1, назвемо її p.
4. Визначажмо, чи є p внутрішньою точкою S_2 (час O(n)). Якщо це не внутрішня точка, то переходимо до кроку 6.
5. Нехай p є внутрішня точка P_2:



Множини P_1 та P_2 впорядковані за зміною кута відносно p. Переходимо до кроку 6.

6. Нехай p не є внутрішньою точкою P_2:



Якщо дивитись із точки p, то многокутник P2 лежить у клині з кутом розвороту, який є меншим або рівним π. Цей клин визначається двома вершинами u та v многокутника P_2, які можуть бути знайдені за лінійний час за один обхід вершин многокутника P_2.

Ці вершини розбивають границю P2 на два ланцюги вершин. Ланцюг, ближчий до p, не буде належати опуклій оболонці, його відкидаємо. Ланцюг, що далі від p, є монотонним за зміною кута відносно p.

7. Маємо дві множини точок P_1 та $P_{2,}$ або P_1 та P_{2outer}, обидва впорядковані за кутом відносно p, за час O(n) їх можна обєднати в одну опуклу оболонку за допомогою обходу Грехема.

Складність алгоритму:

Складність схожа на merge sort — завжди O(n log n).

На кожному кроці робимо O(n) операцій, і розбиваємо множину на дві підмножини, при цьому відомо, що підмножини рівні, тому зробимо рівно. Отже, час виконання цього алгоритму кращий за алгоритм ШвидкОбол, який в найгіршому випадку буде $O(n^2)$.

Існують також інші алгоритми, які знаходять опуклу оболонку за O(n log n) операцій (наприклад, метод Грехема, метод Ендрю), проте алгоритм “Розділяй та володарюй” можна легко пришвидшити за рахунок паралельних обчислень.

Реалізовано на мові Python.

Інтерфейс користувача: набір точок задається файлом points.txt.
Програма у вікні показує всі задані точки та малює опуклу оболонку.