Грищенко Юрій ІПС-32
Лабораторна робота №8

Постановка задачі: Впорядкована множина точок розглядається як простий многокутник. Чи можна побудвати опуклу оболонку простого многокутника за час, кращий за O(N log N)?

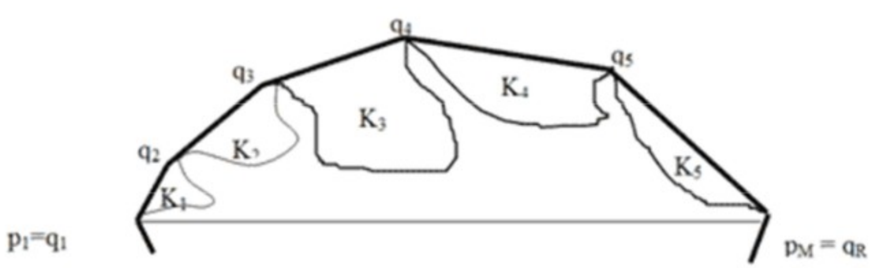
Опис алгоритму (алгоритм Лі)**:**
Нехай p_1 – найлівіша вершина заданого простого многокутника q, а $(p_1, p_2, \ldots p_N)$ - впорядкована циклічна послідовність його вершин.

Нехай внутрішня частина P залишається праворуч при обході його границі у вказаному порядку (тобто множина вершин многокутника орієнтована за годинниковою стрілкою).

Нехай p_M – найправіша вершина. Вершини p_1 та p_M будуть граничними точками опуклої оболонки многокутника q. Вони розбивають послідовність вершин многокутника на два ланцюги: один від p_1 до p_M, другий — від p_M до p_1.

Достатньо дослідити побудову опуклої оболонки для ланцюга $(p_1, \ldots p_M)$ — верхньої оболонки.

Нехай $(q_1, q_2, \ldots q_R)$ — підпослідовність послідовності $(p_1, p_2, \ldots p_M)$, де $q_1 = p_1$ та $q_R = p_M$ — шукана опукла оболонка многокутника.



Кожне ребро q_iq_{i+1} є “кришкою” “кишені”. Алгоритм проходить ланцюг та послідовно будує кришки усіх кишень.

Критичною будемо називати вершину, яка з останньою знайденою вершиною типу q утворює кишеню. Вони є кандидатами у вершини опуклої оболонки.

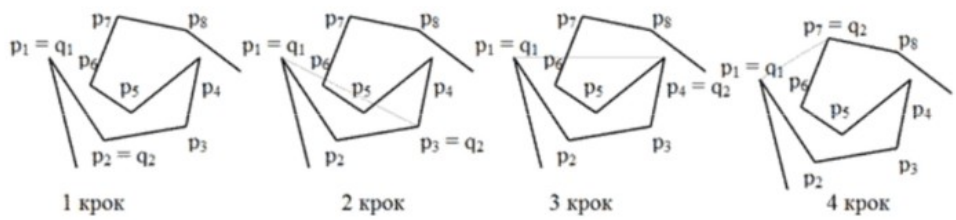
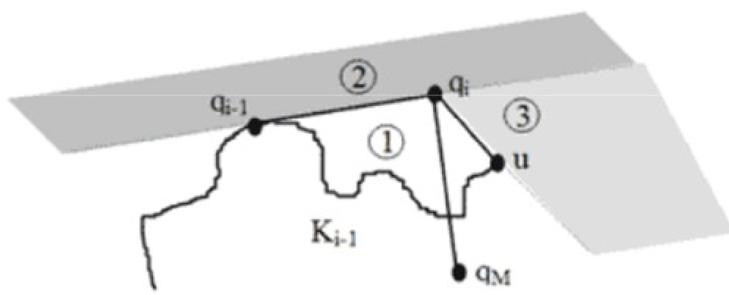


Рисунок 93. Ілюстрація просування по критичним вершинам

Позначимо через u вершину границі P, яка передує q_i.

Залежно від положення u відносно орієнтованого відрізка $p_M q_i$ можливі два випадки:



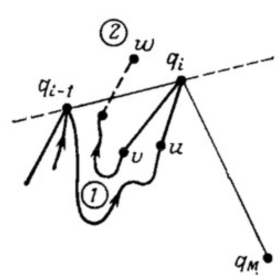
I. Вершина u знаходиться праворуч $q_M q$ або на ньому.
Маємо 3 області, які визначаються:

- прямою, що проходить через точки q_{i-1} та q_i;
- променем, який є продовженням відрізку $q_i u$;
- частиною границі многокутника P, яка відповідає поточній кишені.

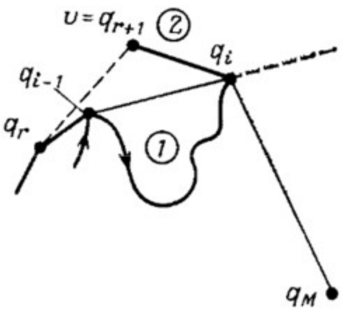
II. Вершина u знаходиться ліворуч $p_M q_i$. У цьому випадку до розгляду додається четверта область.

Позначимо через v вершину, яка слідує за q_i на границі многокутника P. Ця вершина буде знаходитись в одній з вказаних областей розгляду.

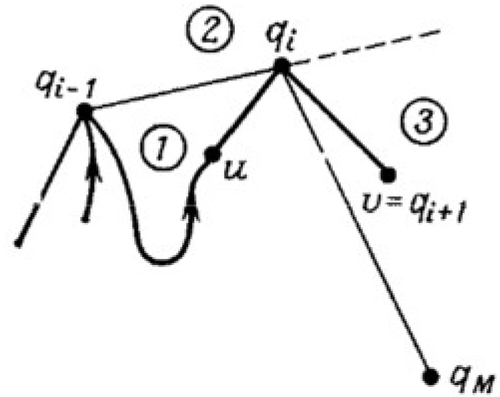
1. Границя заходить в кишеню. Рухаємось по границі до тих пір, поки не досягнемо вершину w, яка знаходиться ззовні кишені.



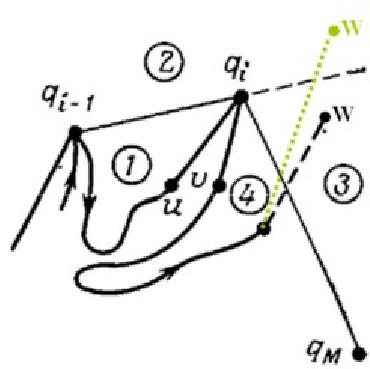
2. Вершина v є критичною. Шукається опорна пряма з вершини v до ланцюга (q_1, q_2, ... q_i). Можливо, будуть вилучені деякі q_{i-1}, q_{i-2}, ... Тоді v — вершина опуклої оболонки.



3. Вершина v є критичною. Тоді v — вершина опуклої оболонки.



4. Границя заходить всередину опуклої оболонки. Рухаємось по границі многокутника доти, поки не досягнемо першої вершини, яка має властивість:
   - належить області 3 або 2, і обробляється відповідно.
   - співпадає з q_M, процедура завершується.



Складність алгоритму:

Після ініціалізації кожна вершина границі відвідується рівно один раз, перш ніж вона буде прийнята або виштовхнута. Обробка кожної вершини многокутника здійснюється за константний час. Послідовності (p_1, ..., p_M) та (q_1, ..., q_R) містять O(M) – елементів.

Нижня оболонка будується аналогічно.

Отже, опукла оболонка простого многокутника з N вершинами може бути побудована за оптимальний час Θ(N) при використанні пам’яті об’ємом Θ(N). Це швидше за задачу пошуку опуклої оболонки в загальному випадку (нижня межа — Ω(N log N))

Реалізовано на мові Python.

Інтерфейс користувача:

Впорядкований набір точок простого прямокутника задається файлом points.txt. Програма у вікні показіє многокутник та опуклу оболонку (як верхню, так і нижню). У консоль виводиться debug-інформація.