CONVEX HULL

Honors Project

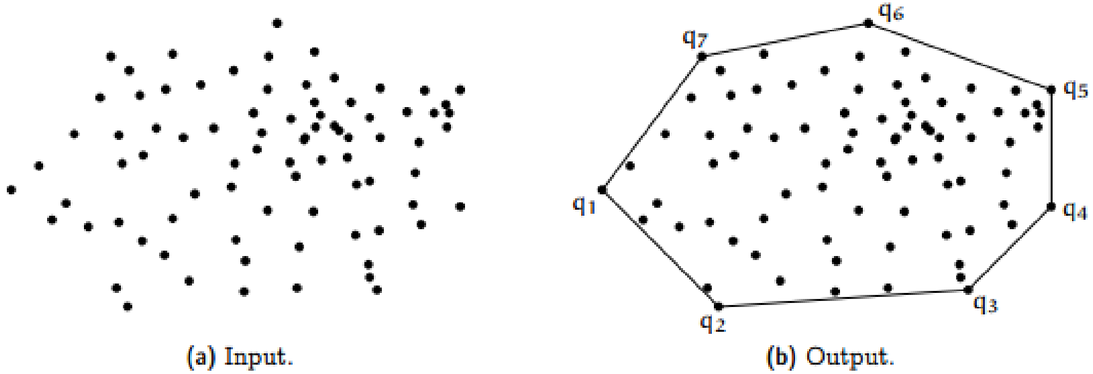
Presented by Saish Gondkar

Problem Statement

Given a set of points on a 2D plane, the Convex Hull is the smallest polygon that encloses all the points.

Input: Set of n points (x,y)

Output: A polygon formed by outerpoints



(a) Input.

(b) Output.

Applications

Real-world uses of Convex Hull:

- Computer Graphics: shape boundaries, object recognition
- Robotics: obstacle avoidance & path planning
- GIS / Mapping: boundary detection of cities or islands
- Machine Learning: clustering shape boundaries
- Physics Simulations: collision detection

Algorithm Overview (Graham's Scan)

Steps of the algorithm:

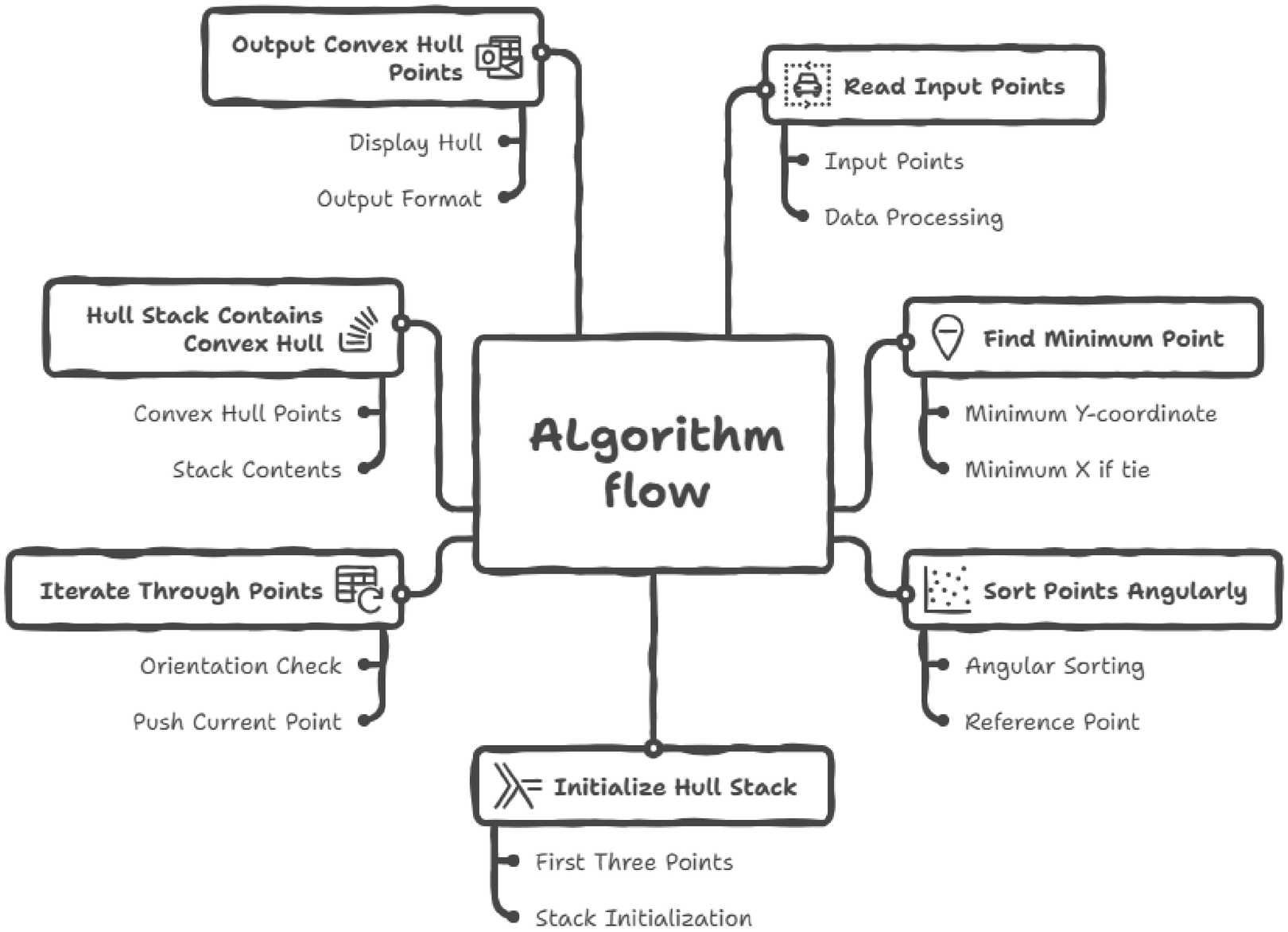
1. Find the lowest (pivot) point
2. Sort remaining points by polar angle
3. Remove collinear points
4. Traverse sorted list and maintain stack of hull points

Code Structure

Files in the project:

- main.c → Controls flow
- io.c/io.h → Reads points from file
- convex_hull.c/h → Algorithm implementation
- README.txt → How to compile & run
- points_10.txt / points_50.txt / points_1000.txt / points_special.txt → Input files

Graham's Scan Algorithm Flowchart



Time Complexity Derivations

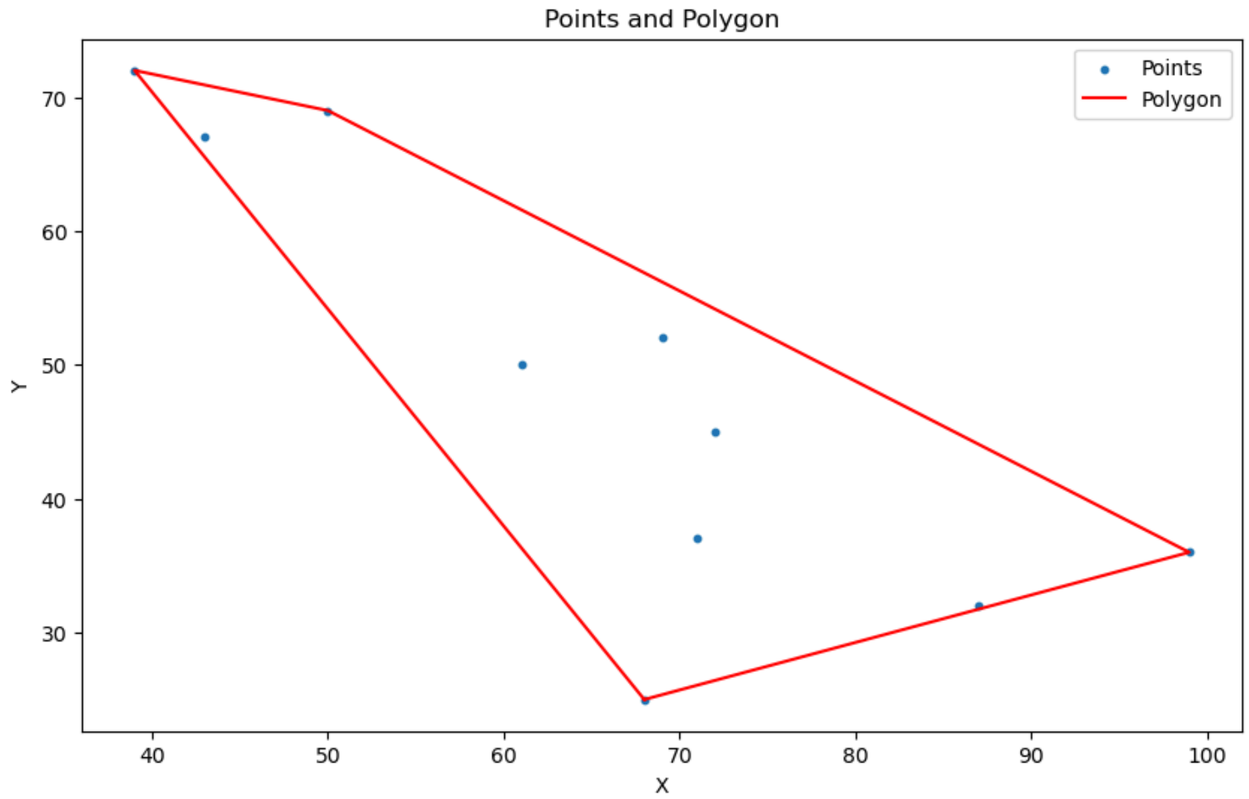
Step	**Description**	**Time Complexity**
Finding pivot	Linear scan	O(n)
Sorting by Polar Angle	Comparison sort	O(n log n)
Removing collinear points	Single pass	O(n)
Building hull (Stack)	Push/Pop once per point	O(n)

Total Time Complexity:

$O(n) + O(n \log n) + O(n) + O(n) = O(n \log n)$

Result

X	Y
68	25
99	36
39	72
87	32
43	67
61	50
72	45
71	37
50	69
69	52



Lessons Learned

- Importance of algorithm visualization before coding
- Using 2D cross product, Squared Euclidean Distance
- How sorting order affects hull shape
- Learning about qsort_s
- Realized the power of computational geometry in real-world scenarios

Thank You