**Terms:**

**Convex Hull:**

A convex hull is the smallest convex polygon that contains a given set of points.

**Convex Polygon**

A convex polygon is a polygon in which all interior angles are less than 180 degrees.

**Collinear Points**

Points that lie on the same straight line.

**Turn or Orientation**

Describes the direction of a turn when traveling from one point to another. It can be:

* Clockwise (CW): A right turn.
* Counterclockwise (CCW): A left turn.
* Collinear: No turn (points are in a straight line).

**Convexity**

A property of a shape where every line segment connecting two points inside the shape stays entirely within the shape.

**Hull Points**

The points that form the boundary of the convex hull. These are the "outermost" points of the set

**Extreme Points**

The points in a set that are farthest in any direction, like the leftmost, rightmost, topmost, or bottommost points. These points are always part of the convex hull because they define its outer boundary.

**Polar Angle**

The angle a point makes with a reference line (usually the x-axis) and a fixed point (like the origin). Used to sort points in algorithms like Graham's Scan.

**Sweep Line**

An imaginary line that moves across a set of points (horizontally, vertically, or otherwise) to systematically process points in algorithms.

**Hull Edges**

The straight-line segments that connect the hull points, forming the boundary of the convex hull.

**Graham’s Scan**

Logic: Sort points by polar angle and iteratively construct the hull.

Time Complexity: O(n\*logn).

Steps:

* Find the Bottom-Left Point: Identify the point with the smallest y-coordinate (and x-coordinate as a tiebreaker).
* Sort Points by Polar Angle: Sort all other points counterclockwise around the pivot (using the slope or atan2).
* Iteratively Build the Hull:
* Start with the first two points.
* For each new point, check the turn direction with the last two points in the hull.
* If it forms a non-counterclockwise turn, remove the last point from the hull.
* Add the new point to the hull.
* Output the Convex Hull.

**Jarvis March (Gift Wrapping)**

Logic: Iteratively find the next hull point by wrapping around the points.

Time Complexity: O(n\*h), where h is the number of hull points.

Steps:

* Start with the Leftmost Point: Find the point with the smallest x-coordinate.
* Iteratively Find Hull Points:
* From the current point, find the point that makes the smallest clockwise angle with the previous segment (use cross product).
* Add the selected point to the hull.
* Repeat Until Closure: Continue until the starting point is reached.

**Monotone Chain (Andrew’s Algorithm)**

Logic: Sort points and construct the hull in two passes (lower and upper hulls).

Time Complexity: O(n\*logn).

Steps:

* Sort Points: Sort all points by x-coordinate (and y-coordinate for ties).
* Build the Lower Hull:
* Start with an empty list.
* For each sorted point, add it to the hull.
* While the last two points and the current point form a non-counterclockwise turn, remove the second-to-last point.
* Build the Upper Hull:
* Repeat the above process in reverse order (right to left).
* Combine Hulls: Concatenate the lower and upper hulls (excluding duplicate endpoints).

**QuickHull**

Logic: Divide and conquer by finding extreme points and recursively solving subproblems.

Time Complexity: O(n\*logn) (average), O(n\*n) (worst case).

Steps:

* Find the Extreme Points: Identify the leftmost and rightmost points. These form the initial line segment.
* Partition Points:
* Divide the remaining points into two subsets: points above and below the line.
* Recursively Find Hull Points:
* For each subset, find the farthest point from the line segment.
* Use this point to divide the subset further.
* Continue recursively until no points remain.
* Combine Results: Merge the hull points from all recursive steps.

**Chan’s Algorithm**

Logic: Combines Graham's Scan and Jarvis March to achieve O(n\*logh) complexity.

Time Complexity: O(n\*logh), where h is the number of hull points.

Steps:

* Divide Points into Buckets: Partition the points into groups of size m=⌈n/h⌉.
* Compute Hull for Each Bucket: Use Graham's Scan or any O(m\*logm) method to compute the convex hull of each group.
* Merge Hulls Using Jarvis March:
* Initialize the leftmost point as the starting point.
* Iteratively find the next hull point across all buckets by checking the smallest clockwise angle.
* Repeat Until Closure: Continue finding points until the hull wraps around.