



# CONVEX HULL ALGORITHMS

( GRAHAM SCAN - JARVIS MARCH - QUICKHULL )

Vicky H., Rykir E., Ashley F.

Get Started



# Introduction to Convex Hull

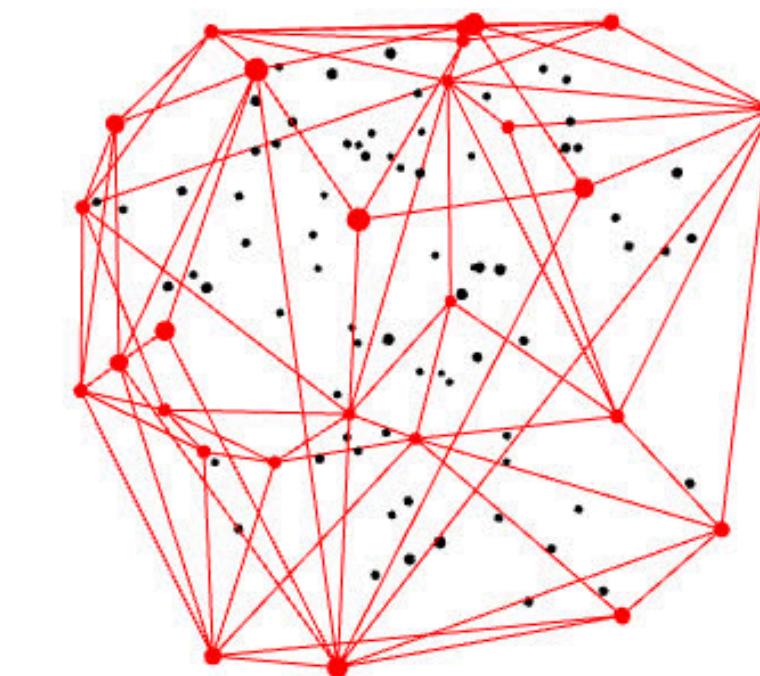
- What is a convex hull?
- Why it matters in computational geometry
- Real-world applications





# Convex Hull

- Convex Shape - Having only interior angles less than  $180^\circ$
- The smallest set that encloses all vertices in a graph, forming a convex polygon
- Can be expanded to 3D (polyhedron)





# Neat Facts

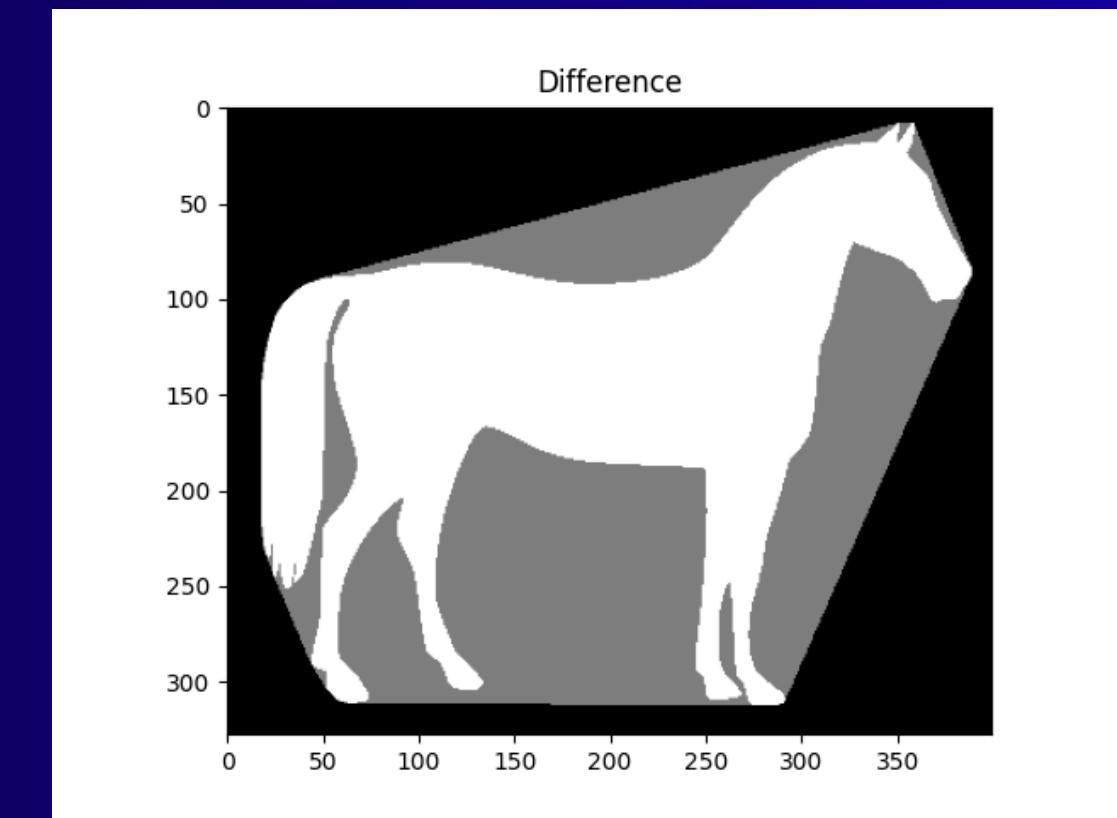
- Convex hulls are unique for each set
- Often analogized to rubber bands stretching around points
- In 2D, Convex Hulls are always polygons.





# Why It Matters

- **Simplicity/Efficiency**
  - Shape estimation
  - Reduced rendering intensity
- **Real World Applications**
  - Pattern recognition
  - Image processing
  - Collision detection (physics and games)





# Convex Hull Algorithms

## Graham Scan

- Published by Ronald Graham 1972
- Time Complexity
  - $O(n \log n)$
  - $n$  = vertices
- Greedy-ish Style

## Jarvis March

- Published by Ray Jarvis in 1973
- Time Complexity
  - $O(nh)$
  - $n$  = vertices
  - $h$  = vertices in hull
- Brute-force-ish Style

## QuickHull

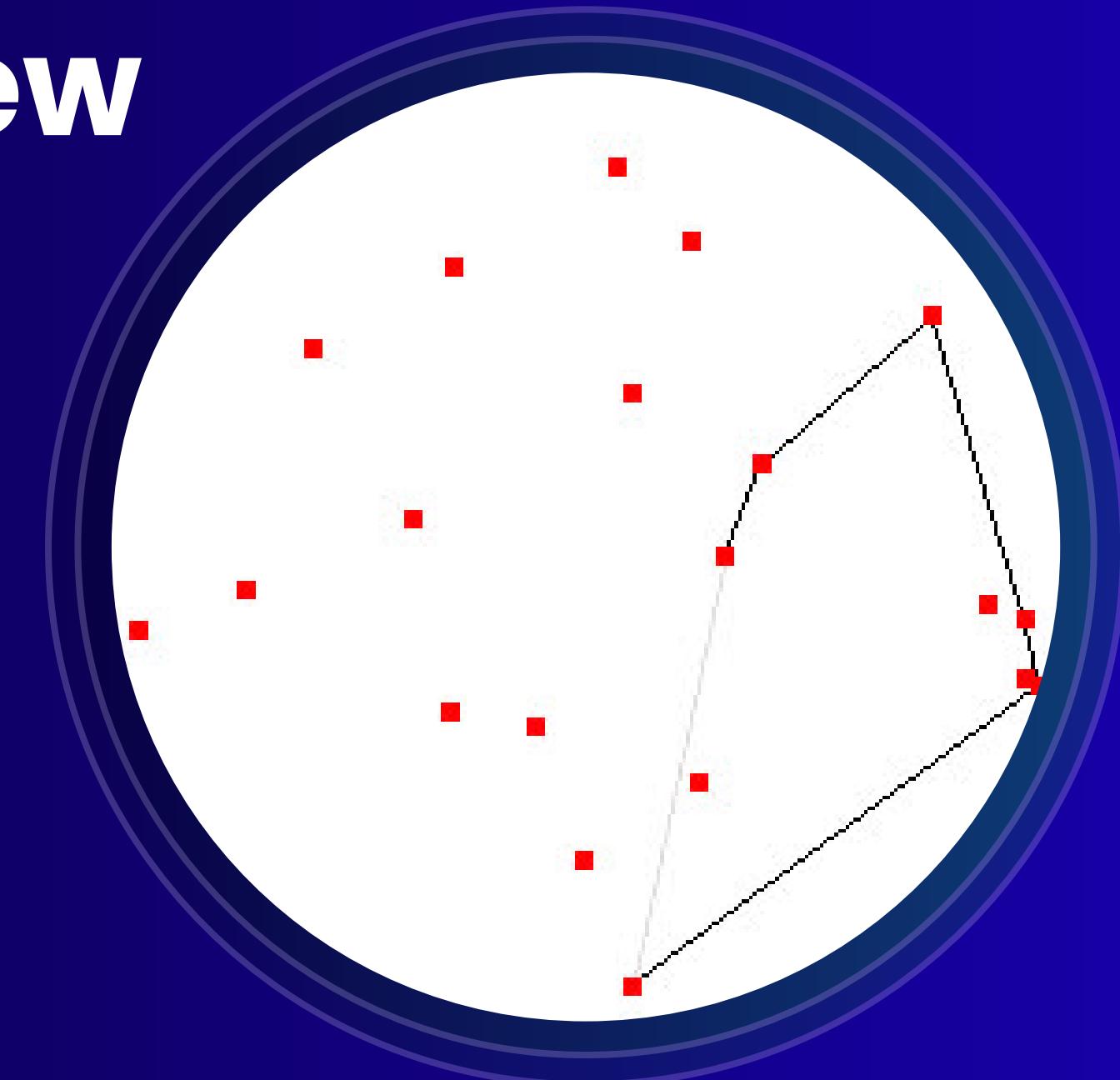
- Published by C. Bradford Barber et. al in 1996
- Time Complexity
  - $O(n^2)$  Worst case
  - $O(n \log n)$  Average
  - $n$  = vertices
- Divide and Conquer



# Graham Scan – Overview

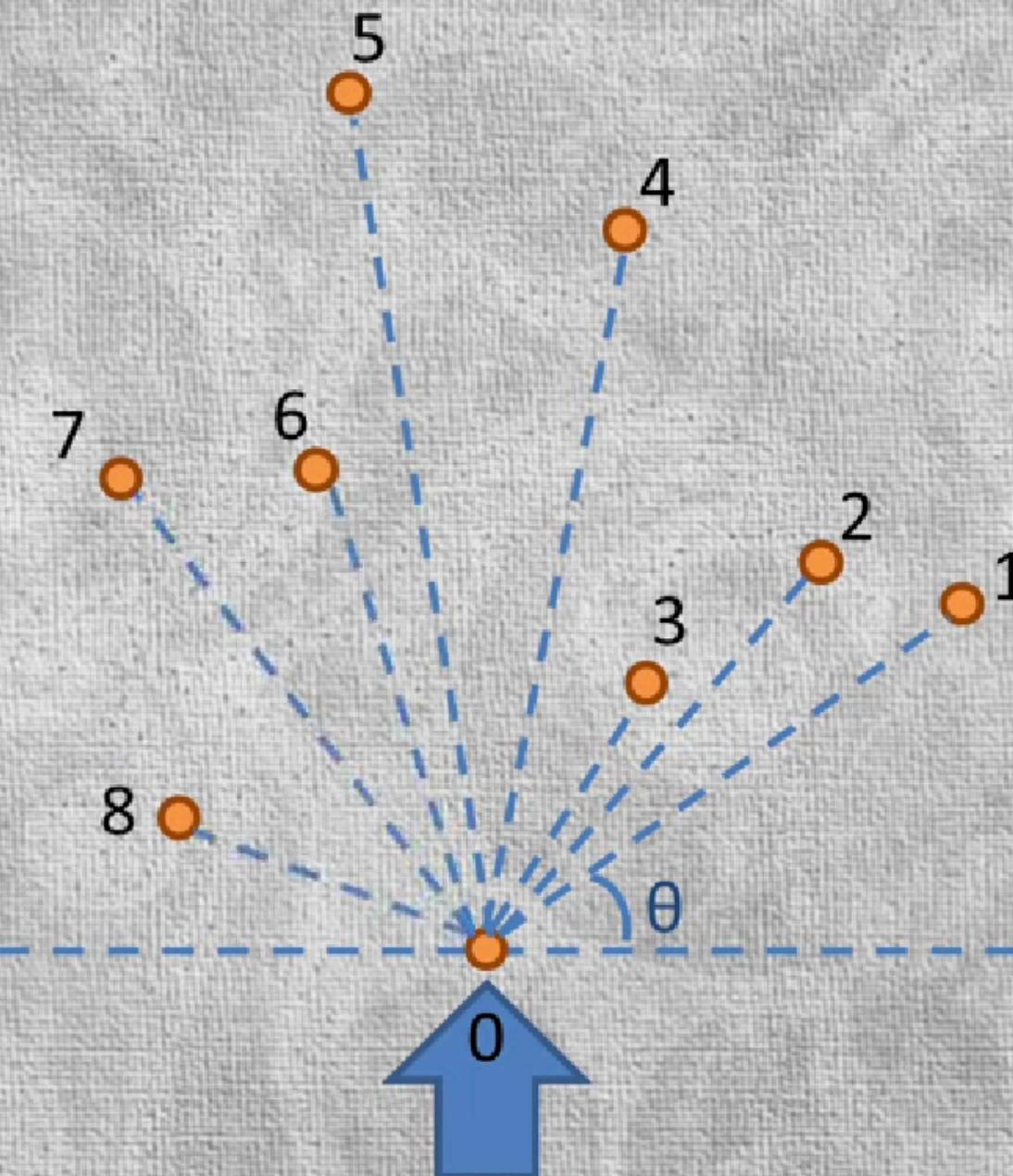
## Steps:

1. Find the lowest point (pivot)
2. Sort points by angle from pivot
3. Use stack to add in-order vertices into the convex hull
4. Keep if the angle is counter-clockwise, remove if clockwise
5. Repeat 3-4 until all points processed





## Graham scan algorithm



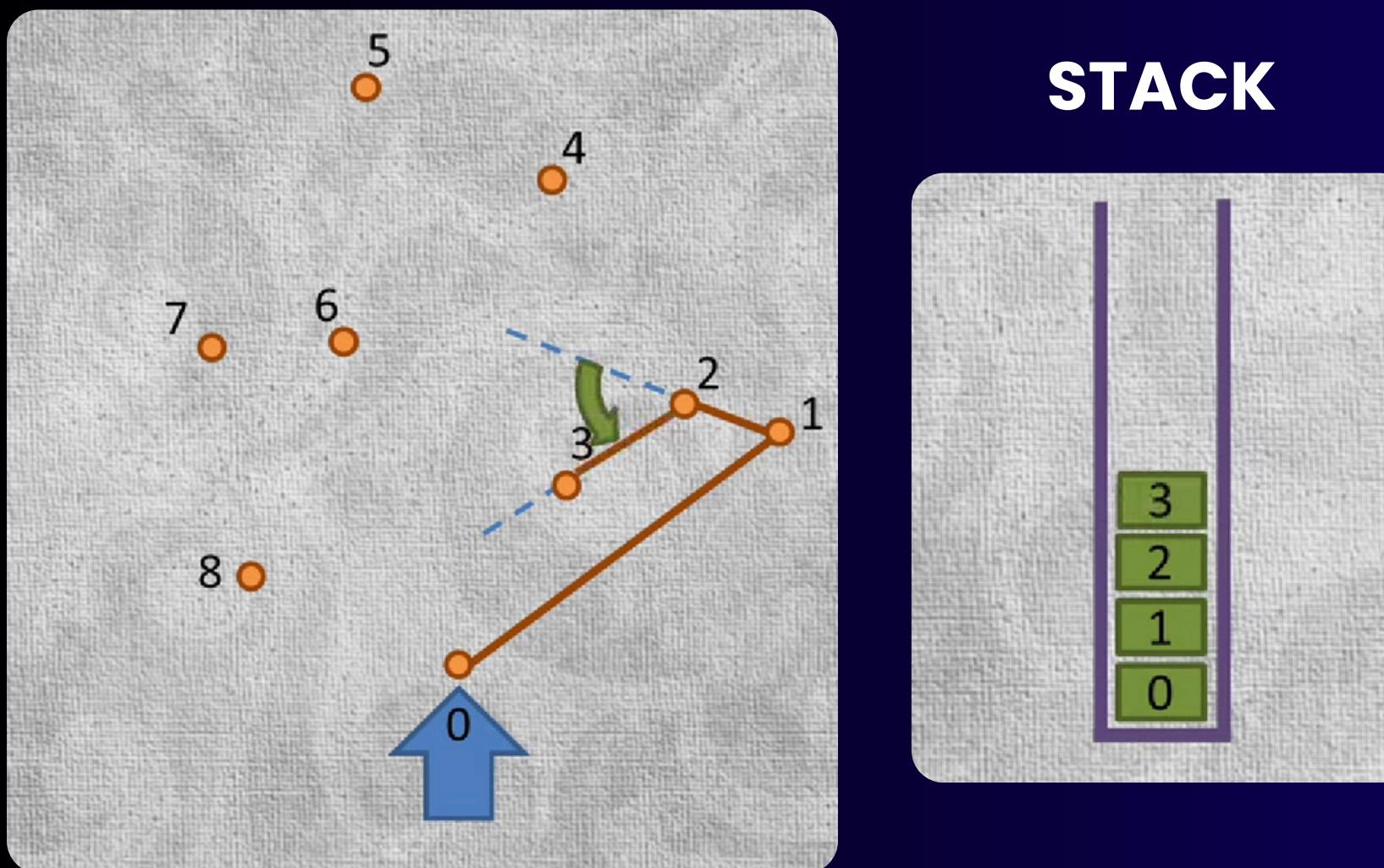
# Point Selection and Ordering

## Steps 1 and 2:

- Iterate over each point
- Declare the point with the smallest y-value as your pivot
- Order points based on the smallest angle made with the pivot



# Point Inclusion and Hull Creation



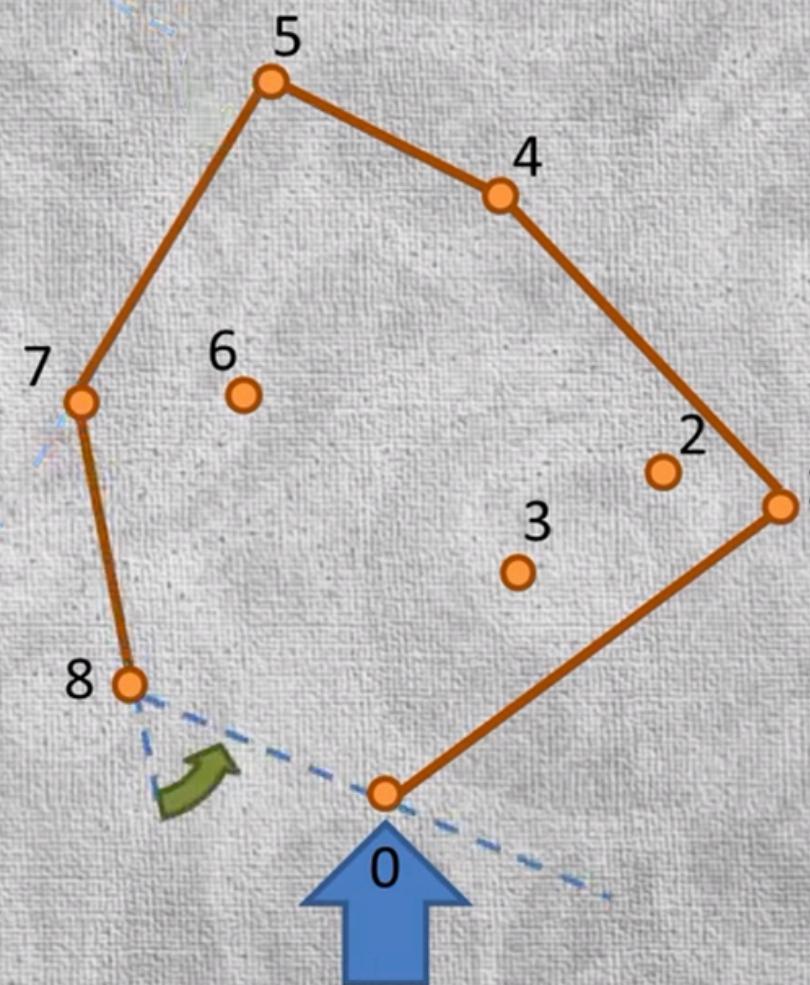
## Steps 3 and 4:

- Add in-order point to stack
- Determine if angle to next point creates a clockwise turn (negative angle) or counter-clockwise turn (positive angle)
- if clockwise, remove from stack and go back
- if counter-clockwise, proceed to next point

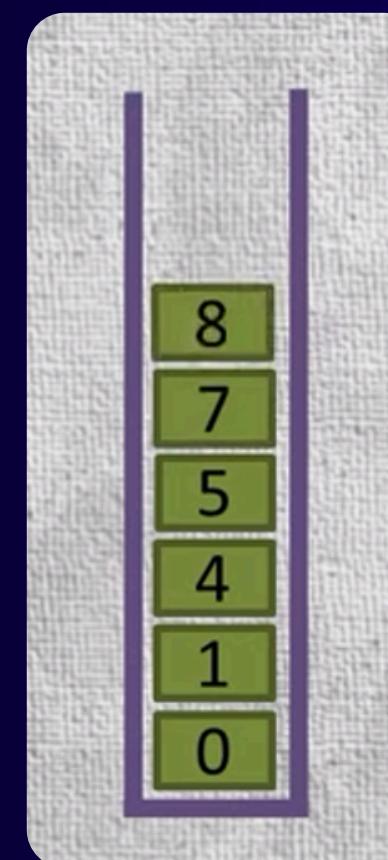


# Conclusion

Graham scan algorithm



STACK



## Step 5:

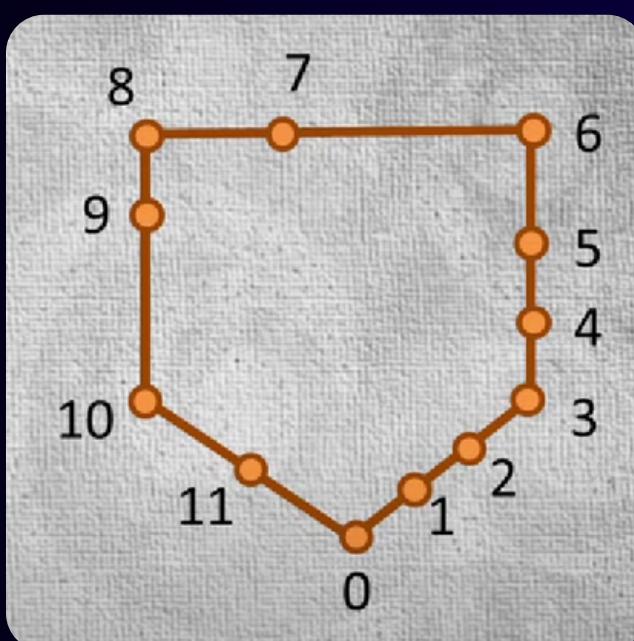
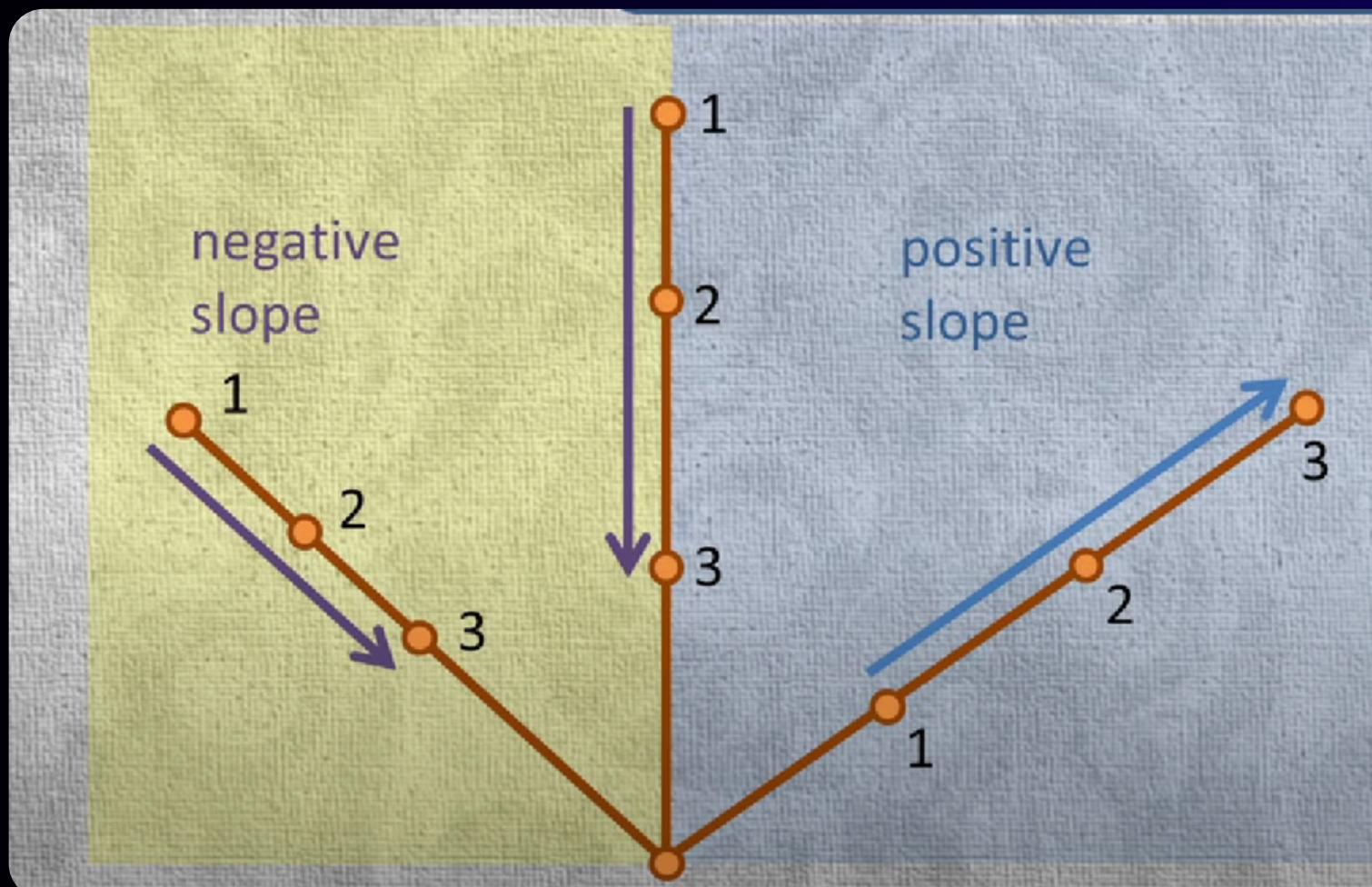
- Once starting point is reached, the algorithm is complete
- The stack contains the points included in the convex hull



# Pitfalls & Edge Cases

## Colinear Points

- Points in a positive slope from start are ordered higher as distance increases
- Points vertical or in a negative slope from start are ordered lower as distance increases

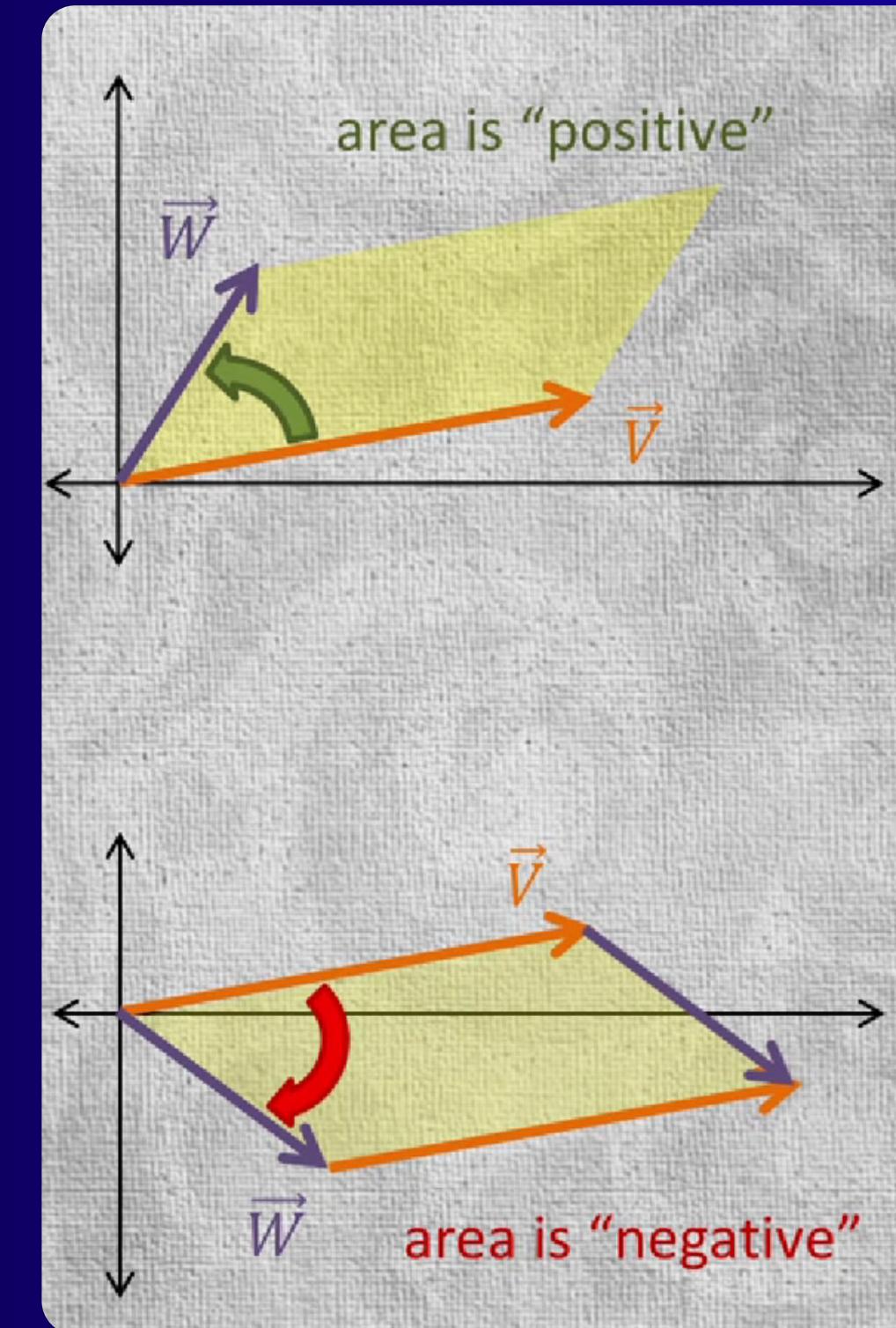




# Implementation:

**How do we know if a turn is Clockwise or counter-clockwise?**

- Calculate Cross product
  - Area is positive if V is on the right of W which forms a clockwise turn.
  - Area is negative if V is on the left of W which forms a counter-clockwise turn.

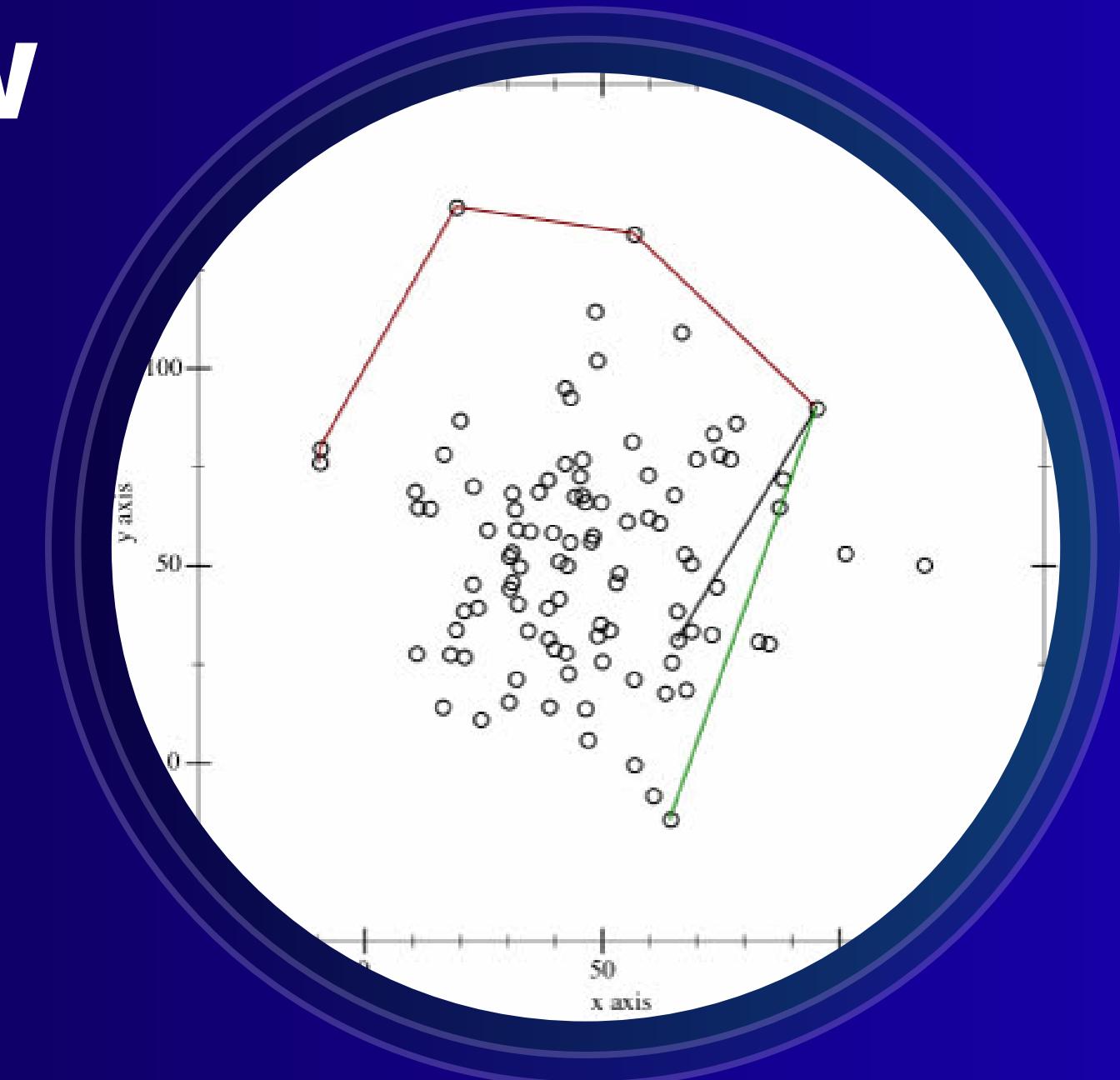


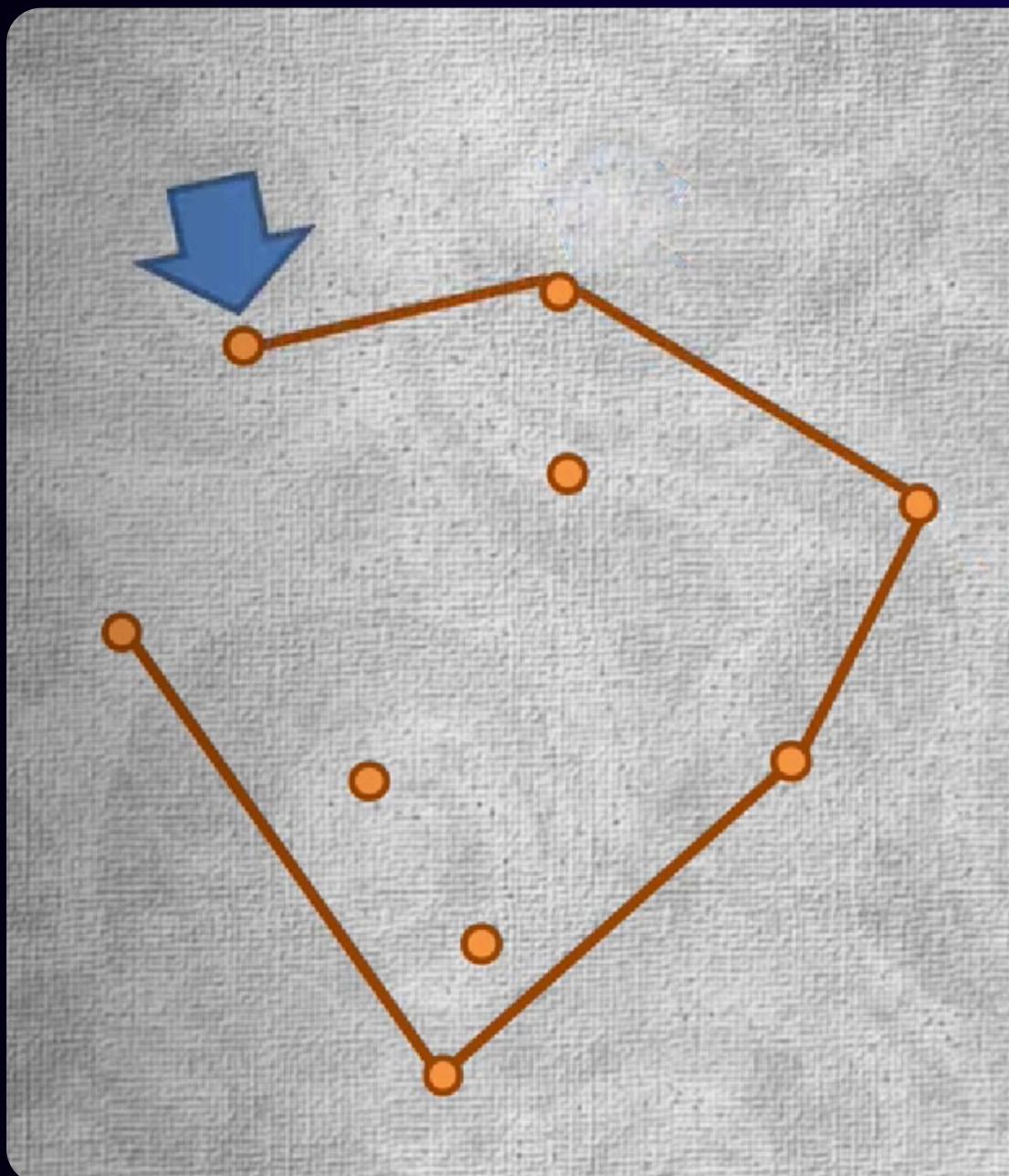


# Jarvis March - Overview

## Steps:

1. Start with the leftmost point
2. From the current point, include the next point such that the line between these two is “outside” of every other point
3. Repeat until you return to the starting point.





# Point Selection

## Step 1:

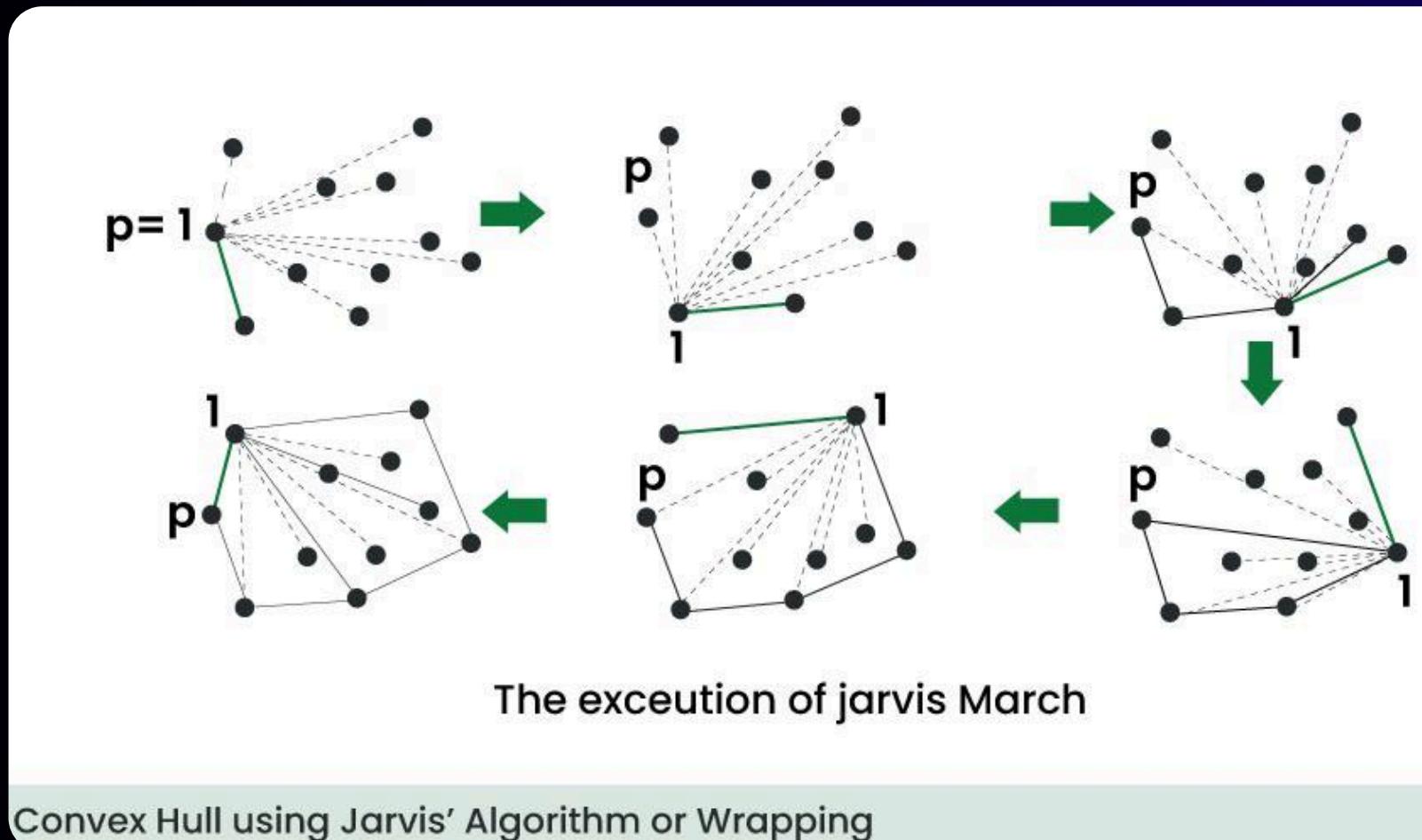
- Iterate over each point
- Declare the point with the smallest x-value as your pivot



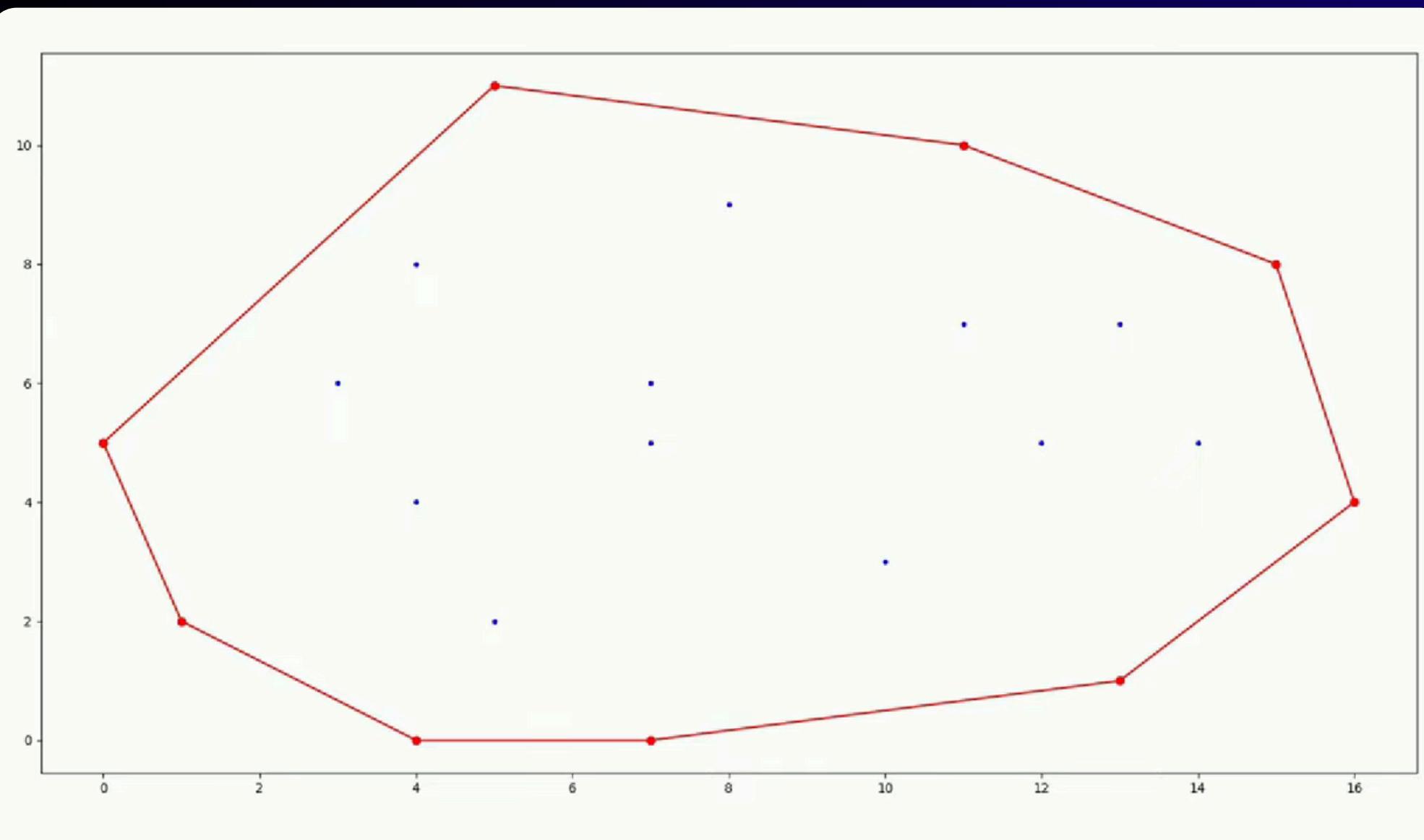
# Main Algorithm

## Step 2:

- From the current point, check every other point
- Select the point with the largest angle to connect and insert it into resultant array
- Draw the connecting line
- This line must be outside of all other points



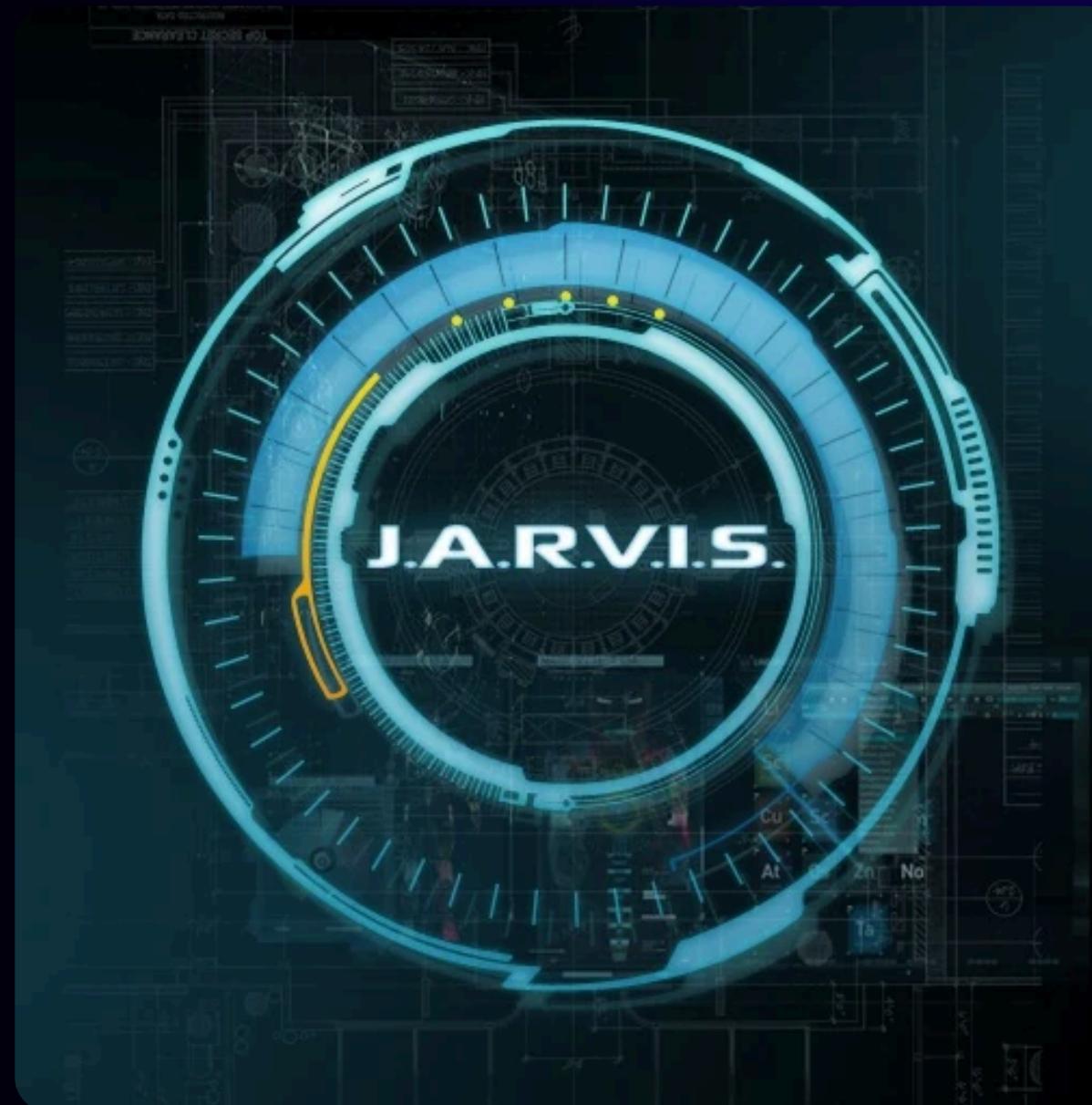
Convex Hull using Jarvis' Algorithm or Wrapping



# Conclusion

## Step 3:

- Once starting point is reached, the algorithm is complete



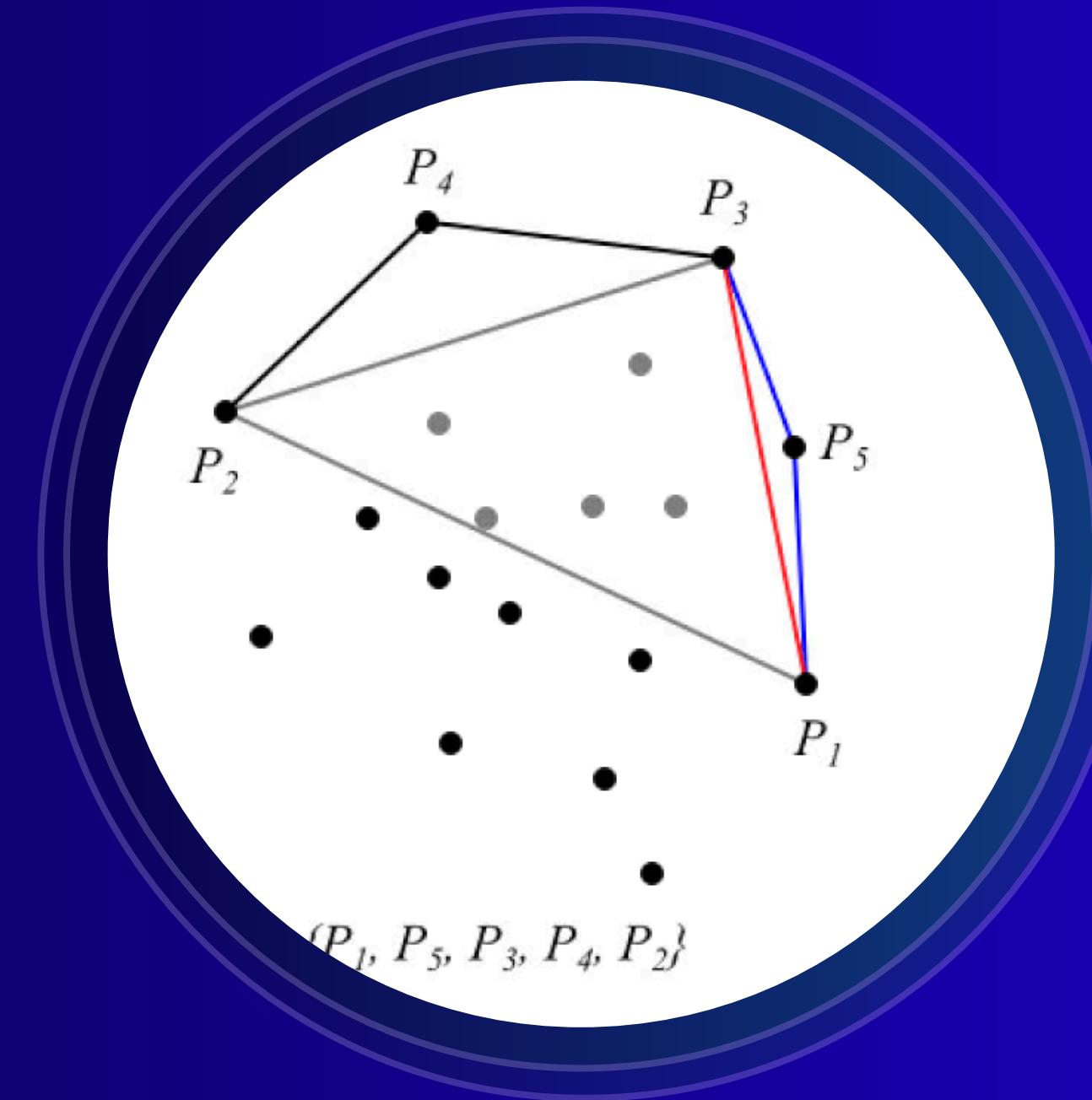
## Cons of Jarvis March

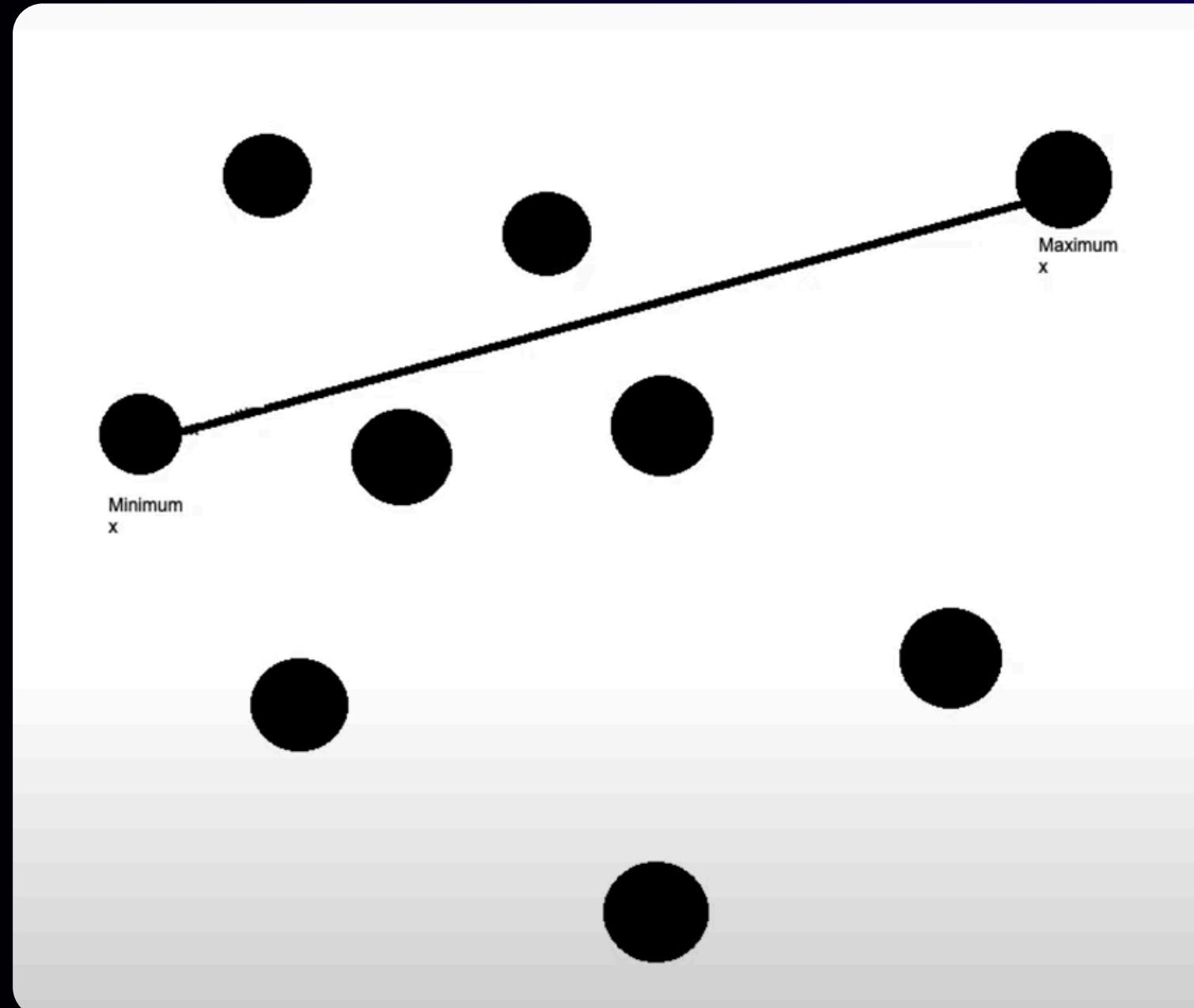
1. Inefficient for Large Datasets
2. Slow Compared to Other Algorithms
3. Harder to Optimize
4. Angle Comparison Is Costly
5. Sensitive to Floating Point Errors



# Quickhull – Overview

1. Find the point with minimum and maximum x-coordinate.
2. Make a line joining the two points.
3. For a part, find the point with the maximum distance from the line. P, min, and max x-coordinates form a triangle.
4. This step splits the problem into two smaller parts, which are solved one at a time.
5. We treat the lines as new edges, look for points outside the triangle, and repeat.

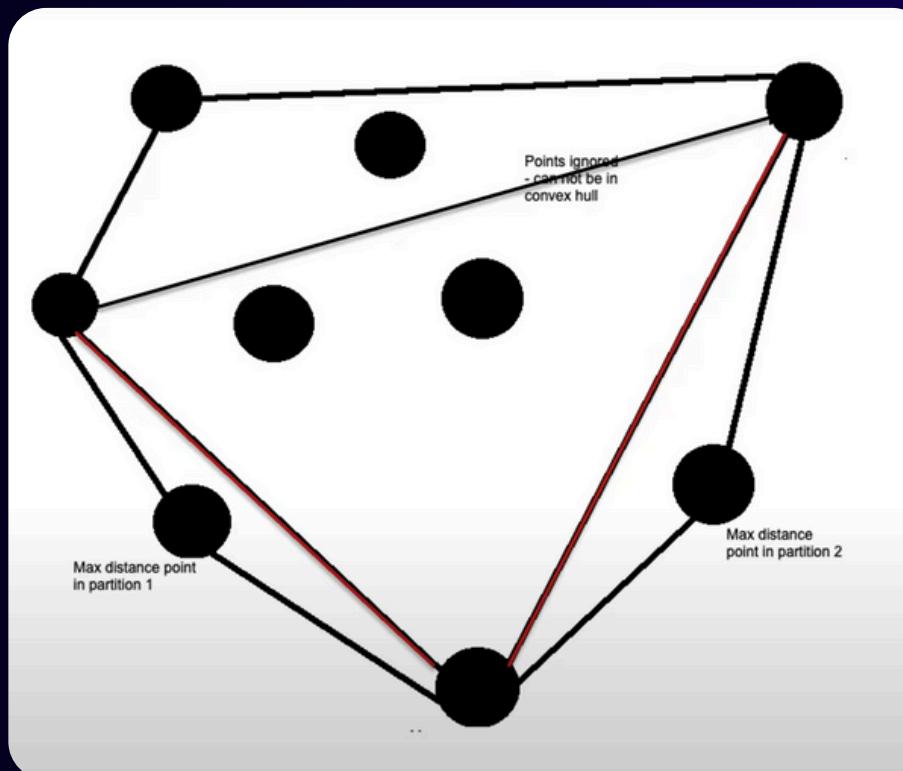
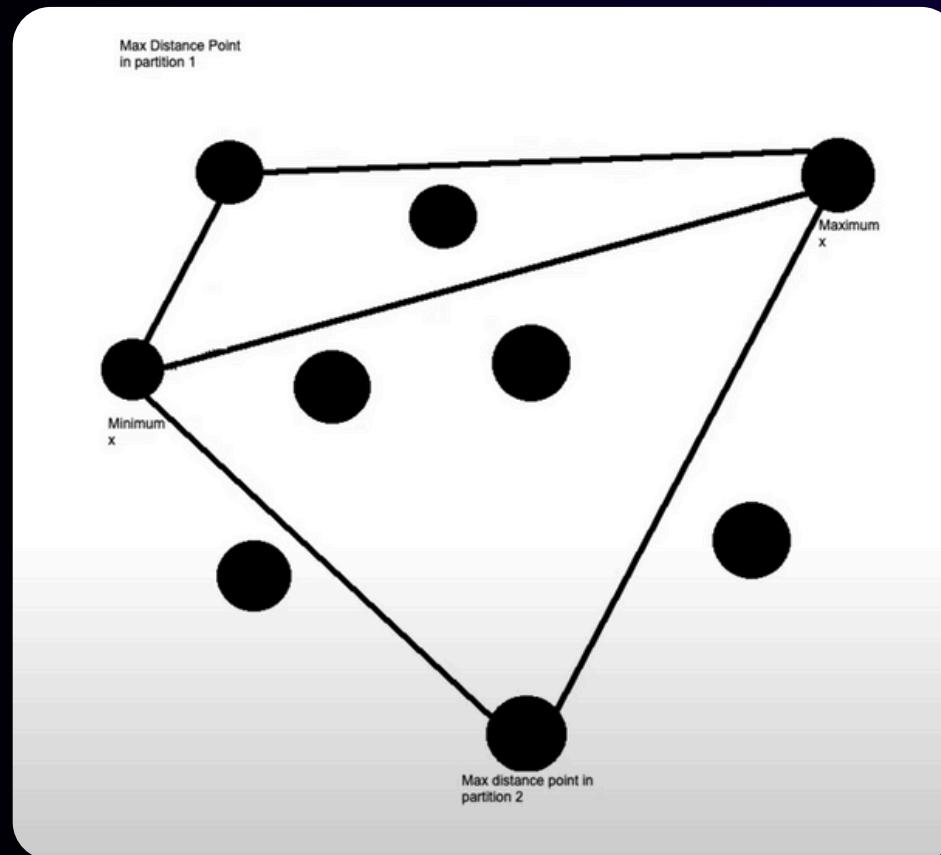




# Point Selection

## Steps 1 and 2:

- Find the point with minimum x-coordinate and the maximum x-coordinate. ( $\text{min}_x$  &  $\text{max}_x$ )
- These two points form a line segment, splitting the set into two subsets: Points above the line and points below the line



# Point Inclusion

## Steps 3 and 4:

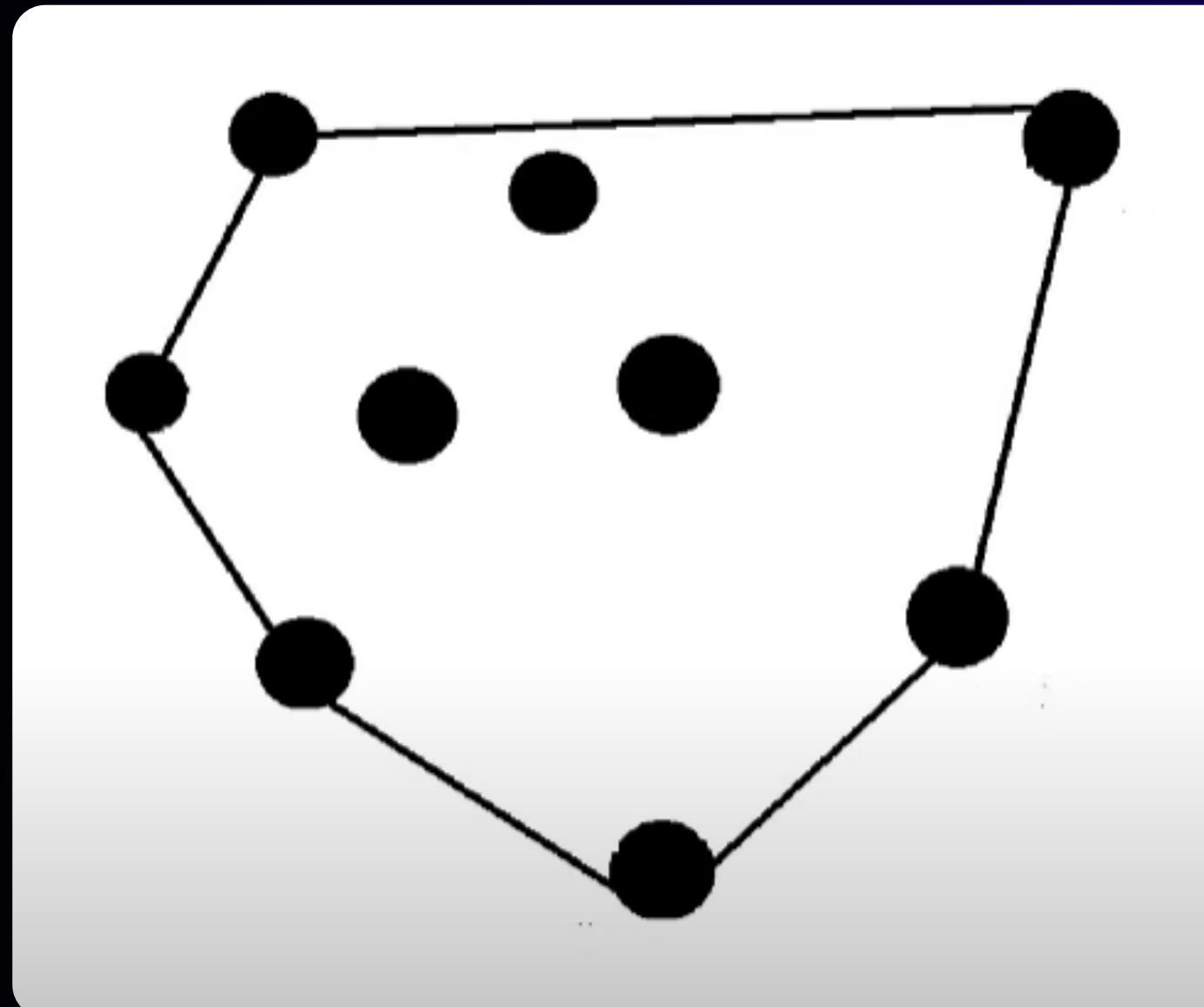
- Recursively find the farthest point from the line to form a triangle.
- Remove points inside this triangle (since they can't be on the hull).
- Recurse on the new edges formed to find more hull points.

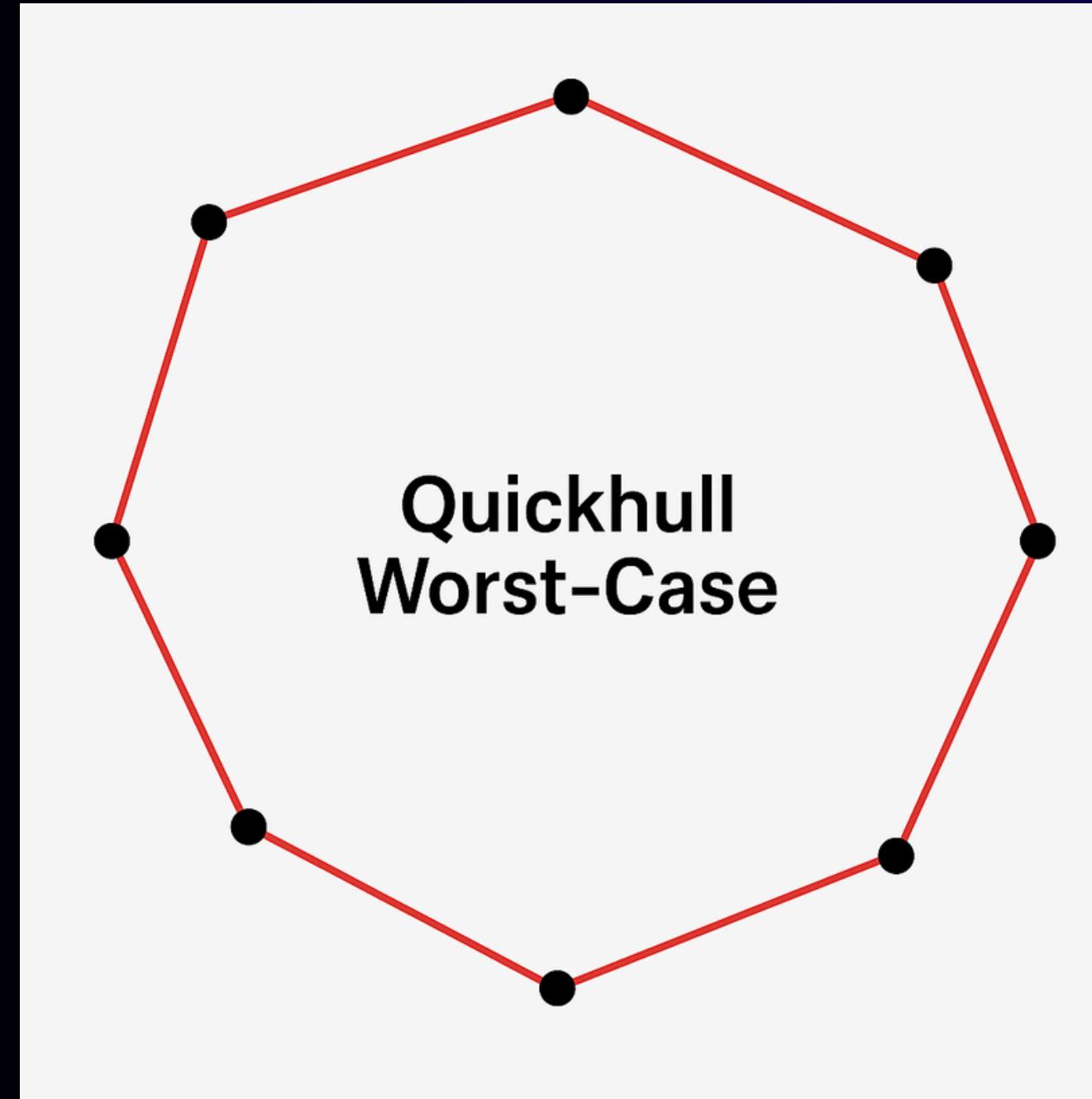


# Hull Creation

## Step 5:

- Continue until no points remain outside the hull edges. The points you've collected form the convex hull.





## Worst Case scenario

The worst-case time complexity of Quickhull is  $O(n^2)$ , and this happens when:

- All or nearly all points lie on the convex hull
- Or, the input points are arranged in such a way that each recursive call removes very few or no points, causing deep recursion and lots of redundant work



# Comparison

ALGORITHM	TIME COMPLEXITY	BEST USE CASE
Graham Scan	$O(n \log n)$	Static, sorted data
Jarvis March	$O(nh)$	Small output size
Quickhull	$O(n \log n)$ avg	Large datasets, fast

# Our Best Team

3013-ALGORITHM

Home

About

Contact



Vicky Heredia



Rykir Evans



Ashley Flores

THANK YOU  
QUESTIONS?



Slide 69

## Sources

- [Convex Hull Algorithms Graham Scan and Jarvis March YouTube Video] (<https://www.youtube.com/watch?v=B2AJoQSZf4M>)
- [Jarvis March Algorithm YouTube Video] (<https://www.youtube.com/watch?v=nBvCZi34Fo&t=329s>)
- [Quick Hull Algorithm Youtube Video] (<https://www.youtube.com/watch?v=s39TMgSvRP0>)
- [Quickhull Algorithm YouTub Video] (<https://www.youtube.com/watch?v=2EKIZrimeuk>)
- [Understanding Convex Hull Algorithms: A Comprehensive Guide] (<https://algocademy.com/blog/understanding-convex-hull-algorithms-a-comprehensive-guide/>)
- [QuickHull Algorithm GeeksforGeeks] (<https://www.geeksforgeeks.org/quickhull-algorithm-convex-hull/>)