Graham Scan Algorithm

MATH 261 Computational Geometry

Input: a set S of n points in the plane, specified by xy-coordinates

Output: a list L of vertices of $\text{conv}(S)$ in counterclockwise order

Algorithm:

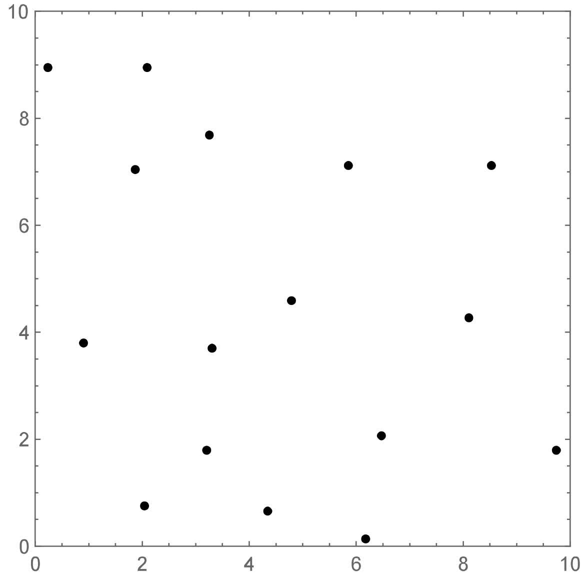
```
anchor = point with lowest $y$-coordinate
sorted = other points, sorted by their angle from anchor

hull = {anchor}
for i = 1 to length(sorted):
  append sorted[i] to hull
  while the next-to-last vertex of hull forms a right turn:
    remove the next-to-last vertex from hull

return hull
```

Answer the following questions:

1. For the following configuration of points, how many points does the algorithm add to the convex hull, only to remove them later? Indicate the order in which these points are added and removed.



2. What is the computational complexity of the Graham Scan algorithm?

3. What configuration of points would cause worst-case runtime for the Graham Scan algorithm?

4. What degenerate configurations of points could cause problems for a naive implementation of the Graham Scan algorithm? How would you avoid such problems?