**77.Convex-Hall problems**

def orientation(p, q, r):

val = (q[1] - p[1]) \* (r[0] - q[0]) - (q[0] - p[0]) \* (r[1] - q[1])

if val == 0:

return 0

return 1 if val > 0 else -1

def graham\_scan(points):

n = len(points)

if n < 3:

return "Convex hull not possible"

hull = []

l = min(range(n), key=lambda i: points[i])

p = l

while True:

hull.append(points[p])

q = (p + 1) % n

for i in range(n):

if orientation(points[p], points[i], points[q]) == -1:

q = i

p = q

if p == l:

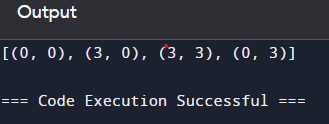
break

return hull

# Example Usage

points = [(0, 3), (2, 2), (1, 1), (2, 1), (3, 0), (0, 0), (3, 3)]

print(graham\_scan(points))



Time complexity:O(n2)