Name: Jigar Siddhpura SAPID: 60004200155

DIV: C/C2 **Branch:** Computer Engineering

AA - Experiment 7 - Convex Hull using Graham Scan

	Jigar Siddhpuna 60004210155
	C22
	a collection of
	Advance Algorithm (AA)
22.50	Experiment 07 · Convex Hull
13.0	clsing Guaham Scan
- 6 1	AIM: To implement convex Hull using OHAHAM Scan
6700	Theory:
	Graham's & can is a method for finding the convex hull of a finite wet of a points in the plane with time complexity
	ogragn where n'is the number of input points. The algorithm
	stank by selecting the point with the lowest y co-ordinate
	as the pivot point . It then start the memaining points by the
	polar angle they make with the pivot point, breaking nes
7-7-16	by delecting the point closet to the pivot . This southing step
	such as quickbank on mergecont.
	Once the points are starled, the algorithm structes through them
	in order, adding each point to the convex him if it makes a
	hul, If a point makes a clock-wise hun, it is not part
	of the cov convex hull and is nemoved.
	This process ensures that the convex hull is computed efficiently by eliminating points that lie within the convex hull.
Gundaram	Finally, the algorithm returns the list of points that form of the convex had in counter- activise order from the pivot point.

	Conclusion:
	Control of the second of the s
	awaham's scan algorithm efficiently computes the convex
	hull go set of points an the plane by souting points based
	on their polar er angles from a pivot point. By grenatively
	adding points that Jarma counter- clockwise hum with the
	last two points on the convex hull, it constructs the convex
	hud in O(nlogn) time.
7 0 114	Therefore we have implemented convex hill graram scan.
	Sample Brain State Short, and all distance to the territy
30 31500	Sull of tilly factor by Johnson School of Storing agency
T. Proj	Carro do u aporto da Africa Andre a de preferancia de Mundea
700 44	A Wind To William A to Street or would be follow townly to the go
- 24	palar and thing and the the state is a state that
	Station and taxis and taxis and taxis and and taxis and
	on the conformation of court within the training that
	duck as quickents with insergers it
14-14-1	and Many making and makes a street and a sure
	The Call and a country was of their term of the country of
~ ~ 133	
6 400	TO A CHEST OF THE STORY OF THE
	to the design of the fact about the part of the fact o
And the	THE COURSE CHANGE HAVE COLORS LIVE IS COLORS TO THE
2 13	a school of the state of the school of
	the principle that are presented under the principle of the control of
- 0.1	on it were a finite on the land with the faction of the a
	· · · · · · · · · · · · · · · · · · ·
Sundaram	FOR EDUCATIONAL USE

CODE:

```
import math
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 # Collinear
  return 1 if val > 0 else 2 # Clockwise or counterclockwise
def graham scan(points):
  n = len(points)
  if n < 3:
    return []
  min point = min(points, key=lambda x: (x[1], x[0]))
  sorted points = sorted(points, key=lambda x: (math.atan2(x[1] - min_point[1], x[0] - min_point[0]), x))
  convex hull stack = [sorted points[0], sorted points[1], sorted points[2]]
  print("Stack after adding first 3 points:", convex hull stack)
  for i in range(3, n):
    while len(convex hull stack) > 1 and orientation(convex hull stack[-2], convex hull stack[-1],
sorted_points[i]) != 2:
      convex hull stack.pop()
    convex hull stack.append(sorted points[i])
    print("Stack after adding point", sorted points[i], ":", convex hull stack)
  return convex hull stack
points = [(0, 3), (2, 2), (1, 1), (4, 4), (1, 2), (3, 1), (0, 0), (3, 3), (1, -1)]
convex hull = graham scan(points)
print("Convex Hull:", convex_hull)
```

OUTPUT:

```
PS D:\SEM-6\AA\EXPERIMENTS> python -u "d:\SEM-6\AA\EXPERIMENTS\ConvexHullProblem.py"
Stack after adding first 3 points: [(1, -1), (3, 1), (4, 4)]
Stack after adding point (3, 3) : [(1, -1), (3, 1), (4, 4), (3, 3)]
Stack after adding point (2, 2) : [(1, -1), (3, 1), (4, 4), (2, 2)]
Stack after adding point (1, 1) : [(1, -1), (3, 1), (4, 4), (1, 1)]
Stack after adding point (1, 2) : [(1, -1), (3, 1), (4, 4), (1, 2)]
Stack after adding point (0, 3) : [(1, -1), (3, 1), (4, 4), (0, 3)]
Stack after adding point (0, 0) : [(1, -1), (3, 1), (4, 4), (0, 3), (0, 0)]
Convex Hull: [(1, -1), (3, 1), (4, 4), (0, 3), (0, 0)]
PS D:\SEM-6\AA\EXPERIMENTS>
```