Experiment No. 3

Aim – Experiment based on divide and conquer approach.

Details –In Computational Geometry, the convex hull or convex envelope or convex closure of a shape is the smallest convex set that contains it. There are many algorithms which can be used for computing 2D convex hulls, namely Graham's Scan Convex Hull and Divide Conquer Convex Hull. The

Brute Force Convex Hull – There will be n^2 pairs of numbers as line segments for n number of 2-D points. Test each line segment to see if it makes up an edge of the convex hull. If the rest of the points are on one side of the segment, the segment is on the convex hull else the segment is not. The complexity of the brute force algorithm is $O(n^2)$ edges, O(n) tests and thus the algorithm complexity is $O(n^3)$.

Graham's Scan Convex Hull – The first phase of the algorithm is to identify the minimal point on some x-axis. Once the minimal x-axis point has been located, and the remaining points sorted in increasing value of x-value of points. The polygon starts as a perimeter to every point in the set. Graham's Scan relies on the convexity of a point, relative to its neighbors, to determine if it is part of the convex hull. Starting at the second node in the perimeter (the first node clockwise after the minimal point), the algorithm systematically tests every point. If the point is convex (i.e. right turn), it proceeds to the next point. If the point is not convex (concave i.e. left turn), it removes the current point from the perimeter list. Rather than progressing as usual however, the algorithm needs to backtrack to the previous point to see if the change has caused other nodes to become convex. This process continues until having looped around the perimeter, back to the minimal point. Once this occurs, the points remaining in the perimeter list are nodes in the convex hull.

Divide and Conquer Convex Hull – All points are sorted by their X coordinates. The tie is broken by ranking points according to their Y coordinate.

- First, all points are sorted in ascending order (according to their x-coordinates). The tie is broken by ranking points according to their Y coordinate.
- Next, the points are divided into two halves S1 and S2. The set of points S1 contains the points to the left of the median, whereas the set S2 contains all the points that are right to the median.
- The convex hulls for the set S1 and S2 are found individually. Assuming the convex hull for S1 is C1, and for S2, it is C2.
- Now, the convex hull C1 and C2 are merged such that the overall convex hull C is returned.

Problem Definition & Assumptions – You need to implement three algorithms namely Brute-force, Graham's Scan and Divide and Conquer for finding a minimum covering of a set of 50 points of 2-D Plane. You may use any data structure for storing these points. Finally, compare three algorithms namely Brute-force, Graham's Scan and Divide and Conquer by finding the running time of the three algorithms using the time high_resolution_clock::now() function or similar C/C++ function.

Important Links:

1. C/C++ Rand function Online library
   https://cplusplus.com/reference/cstdlib/rand/
2. Time required calculation Online library-
   https://en.cppreference.com/w/cpp/chrono/high_resolution_clock/now

---

Input –

1) Each student have to generate a set of 50 points of 2-D Plane using rand() function and use this input to three algorithms namely Brute-force, Graham's Scan and Divide and Conquer.

Output –

1) Set of Points of Convex Hull or edges of Convex Polygon
2) Draw the convex polygon on plain paper or use python code for drawing convex polygon
3) Comment on running time of three convex hull algorithms.