Queens College of CUNY
Department of Computer Science
Design and Analysis of Algorithms (CSCI 323/700)
June 2024

Assignment #5:
"Computational Geometry"
Due: June 12, 2024

Overview:

"Computational Geometry" is an important subarea within Computer Science of both theoretical and practical interest that involves programmatically determining the properties of points and shapes in one, two, or three dimensions. In this assignment, we implement several algorithms for Computational Geometry.

Submissions:

In the Google form, please submit the following:

- Assignment4.py (source code)
- Assignment4.txt (console output)
- Assignment4.png (grid with points and lines)

Your console output should be clearly labeled with the question/task number, description, the actual output, and followed by a blank line to separate it from the next section.

Tasks:

[0] Please consult previous assignments for a program template and structure, general guidelines about best practices, as well as sources of and policies concerning using public code.

[1] Define a function generate_points(n, mn, mx) to make a random list of points in 2D, with each coordinate being between mn and mx.

[2] Define a function line(points, i, j) that finds the line between two points in a list.

See: https://www.geeksforgeeks.org/program-find-line-passing-2-points/

[3] Define a function draw_points(points) to draw the points using a package like matplotlib

See the first posting at https://stackoverflow.com/questions/35363444/plotting-lines-connecting-points

[4] Define a function draw_lines(points) to draw the line segments between successive points

See the successive postings at https://stackoverflow.com/questions/35363444/plotting-lines-connecting-points

[5] Define a function sort_points(points) to sort the points in clockwise (or counterclockwise) order

See https://www.tutorialspoint.com/program-to-sort-given-set-of-cartesian-points-based-on-polar-angles-in-python

[6] Define a function to compute the convex hull of a set of points. You may use any algorithm.

See https://www.geeksforgeeks.org/convex-hull-using-divide-and-conquer-algorithm/

[7] Define a function compute_area(points) to find the area of the convex hull, not the original set.

See https://www.geeksforgeeks.org/area-of-a-polygon-with-given-n-ordered-vertices/

[8] Define a function compute_perimeter(points) to find the perimeter of the convex hull, not the original set.

See https://www.geeksforgeeks.org/equable-shapes/

[9] Define a function is_equable(points) to determine if the shape is equable (area and perimeter are equal)

See https://www.geeksforgeeks.org/equable-shapes/

[10] Define a function closest_pair(points) to determine the closest pair in a set of points

Use brute force, or see https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm/

[11] Define a function farthest_pairs(points) to determine the closest pair in a set of points

Use brute force