CSE355/AMS345 Programming Assignment

Jie Gao*

For this programming assignment you can use your favorite platform and favorite programming language in your implementation. The task is to implement **one** of the following algorithms as listed below.

1. Graham Scan Algorithm for computing convex hull for points in the plane.

   Input: the coordinates of n points.

   Output: a convex polygon representing the convex hull.

2. Triangulation of a simple polygon.

   Input: the vertices of the polygon.

   Output: a triangulation of the polygon.

3. Given a triangulation, use the flip algorithm to turn it into a Delaunay triangulation.

   Input: a set of points and their triangulation.

   Output: a Delaunay triangulation of the same points.

We will provide sample input and sample output files for each algorithm in separate files.

You will get 80% of the total grade if your program assumes no degeneracy, i.e., no three points are on a line, no four points on a circle, etc. To get full marks you will need to handle possible degeneracies in the input.

Submission requires a zipped folder of all source files together with a README file explaining how to run your program. If needed, you may be asked to show a demo of your program to the TA.

Please submit to blackboard the final program and README file by Dec 16th.

In the inputs and outputs below, a set of points, a polygon, and a triangulation may be specified as follows:

- Set of points: the first line shows the number of points. After that, each line shows the $[x, y]$ coordinates of one input point.
  Example:
  3
  $[5, 26]$
  $[76, -23]$
  $[20, 221]$
- Polygon: same as the set of points, with its vertices given in counterclockwise order.

*Department of Computer Science, Stony Brook University, Stony Brook, NY 11794. Email: jgao@cs.sunysb.edu

- Triangulation: this assumes a reference set of points. The points are assigned indices in the following order: the first point has index 1, the second point has index 2, etc.

  The triangles are listed as follows: the first line gives the total number of the triangles in the triangulation. After that, each line is a triple of indices $[i, j, k]$ representing a triangle spanning three vertices with indices i, j, k.

  Example: a (trivial) triangulation of 3 points.
  3
  $[5, 26]$
  $[76, -23]$
  $[20, 221]$
  1
  $[1, 2, 3]$
- For your implementations, you may assume that the number of points $n \leq 100$ and all coordinate values x, y are integers s.t. $x, y \in [-250, 250]$.