Honors Project CSE 2320, Spring 2020

Canvas project submission due Monday April 27, 11:59pm or earlier Project presentation due Wednesday, April 29 (or earlier), during instructor office hours

---

Chapter 33.3 from the textbook, "Introduction to Algorithms", presents two methods for computing the convex hull of a set of points. Given a set of points, *Q*, the convex hull of *Q* is a polygon, *P* such that any point from *Q* is either on *P* or inside it.

- (70 points) Write a program that implements one of these methods: Graham's scan or Jarvi's march. The program must get the input data from a file (either by reading from the file, or by reading from the keyboard combined with redirection of input). If you prefer, you can assume that all the points have integers for their coordinates.
- (10 points) Time complexity formula for all methods in the code with derivations (not just the final formula). It can be written in a separate document or as comments in the code.
- (10 points) Create 4 files with points picked as follows:
  - 10 random points (write a function that does this)
  - 1000 random points (done by the same function that generated the random 10 points from above)
  - hand picked points that include special cases such as ties mentioned in the algorithm you are implementing. For Graham's Scan see ties in lines 1 and 2 of the algorithm, for Jarvi's March see text at the bottom of page 1037.
- The file format is given below:

```
n (number of points)
x1 y1
x2 y2
...
xn yn
```

- (10 points) Plot the points and the polygon either as part of your program (in C/C++) or in another more convenient programming language or environment (Matlab, Python, Word). If you are plotting the points from another environment, have your C/C++ program write BOTH: the solution (the polygon points), and all the points in *Q*, to a file and then have that file be used as the input for the other program.
- The complete project must be submitted to Canvas by the Canvas submission deadline (see top of page).
- **Present your work** to instructor by the demo deadline (see top of page). During the presentation, you will:
  1. demo all components of the project
  2. explain the method used and how it works (on paper)
  3. open the source code and explain which piece of code implements specific parts of the method you are implementing
  4. discuss implementation choices, and
  5. lessons learnt.

IMPORTANT NOTE: MULTIPLE INPUT FILES WILL BE USED TO GRADE THE ASSIGNMENT. YOUR CODE SHOULD WORK WITH ANY INPUT FILE FORMATTED AS SPECIFIED ABOVE.

How to submit

The implementation must be in C or C++ (both functional or object-oriented programming are accepted).

The assignment should be submitted via Canvas. Submit a ZIPPED directory called `honors_assignment.zip` (no other forms of compression accepted, contact the instructor or TA if you do not know how to produce .zip files). The directory should contain source code. Including binaries that work on omega is optional. The submission should also contain a file called README, which should specify precisely:

- Name and UTA ID of the student.
- How the code is structured.
- How to run the code, including very specific compilation instructions, if compilation is needed. Instructions such as "compile using g++" are NOT considered specific.

Grading

The assignment will be graded out of 100 points.

1. **A passing score for this project is 75 points. Note that if your program does not terminate or crashes, it does not pass.**