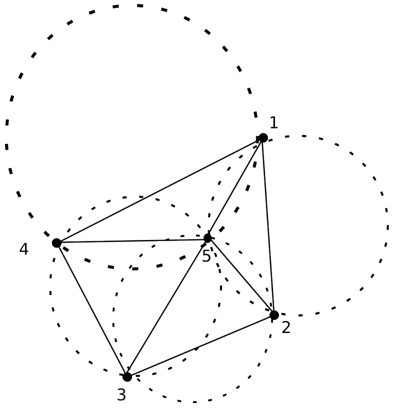
CS558: Homework 3
Due date: Tuesday, March 22, 2011

1 Summary

In this assignment, you will write a program to construct a Delaunay triangulation. As you may recall, a Delaunay triangulation of a finite set of 2D points, $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ is a planar graph, where for each internal face of the graph there is a unique circle which passes through each of its vertices and contains no other vertices. This is illustrated in the following figure:



1.1 Input Format

The input to your program will come from stdin, and consist of an integer $3 \leq n \leq 1000$, representing the number of points, followed by $2n$ floating point numbers representing the coordinates of each point. We shall refer to these points by an index, $1 \leq i \leq n$, which indicates the order of their occurence within the file. Each point will consist of two numbers, $-10^6 \leq x_i, y_i \leq 10^6$ representing its cordinates within the plane. Finally, the input to your program will be in general position; that is no 3 points will be collinear, and no 4 points will be cocircular.

1.2 Output Format

You must output the connectivity information of the Delaunay triangulation as a sequence of edges to stdout. For each edge, j, in the triangulation, write two numbers $s_j < t_j$ on a single line, where s_j and t_j are the indices of the two vertices for edge j. The edges should be written in an order that agrees with the following relation; given two edges, $(s_j, t_j), (s_k, t_k)$;

$$(s_j, t_j) \leq (s_k, t_k) \equiv s_j < s_k \text{ or } (s_j = s_k \text{ and } t_j \leq t_k) \tag{1}$$

1.3 Example Input

```
5
37 74.5
38 62
27 57.5
21.5 66.6
33 67.5
```

1.4 Example Output

```
1 2
1 4
1 5
2 3
2 5
3 4
3 5
4 5
```

1.5 Performance Requirements

Your program must terminate within 10 seconds after starting up. (For timing purposes, both input and output will be redirected from a file, not read from the shell).

2 Written Assignment

1. What data structures do you use to store the triangulation?
2. Describe your algorithm; what is it's time complexity?
3. How would you adapt the output from your code to compute a Voronoi diagram?

3 Grading

You will receive 50% credit for a correct program, and another 50% for a correct analysis and answer to each of the above questions.

4 Extra Credit

There are two things that you can do for extra credit:

1. (+5%) Improve the performance of your program such that it can handle input sizes of up to $3 \leq n \leq 10^6$ points within the time limit.
2. (+5%) Correctly handle all degenerate cases.

5 What to turn in

You should should turn in a gzipped tarball named "*yourname*`.hw3.tar.gz`", where *yourname* is your last name. This archive should contain the following three things:

1. Your source code.
2. A Makefile.
3. A README which contains your answer to the written portion of the assignment.

Email your solution to mikolalysenko@gmail.com with the subject "CS 558 HW3".