CS380 Computational Geometry
Spring 2013

Homework:	**4**	**Professor:**	**Dianna Xu**
Due Date:	**3/19/13**	**E-mail:**	`dxu@cs.brynmawr.edu`
Office:	**Park 246A**	**URL:**	`http://cs.brynmawr.edu/cs380-01`

Assignment 4

You have the choice of doing the theory question (5) or the implementation (6). If you are implementing, submit a printed copy of your code, together with at least three test cases and printouts of sample runs on those test cases. The test cases should be chosen to clearly demonstrate that your code works. You may assume that the input points lie in general position.

1. Exercise 3.3 in O'Rourke

2. Exercise 3.17 in O'Rourke, by "this number" he means the one stated in Theorem 3.16.

3. Exercise 3.21 (a) and 3.26 (a) in O'Rourke

4. Exercise 3.52 in O'rourke

5. Theory Choice In computer graphics, triangulations are used to represent surfaces. Graphics libraries like OpenGL support a special type of triangulation called a *strip*, which is defined as follows. For $n \geq 3$, given a sequence of vertices $V = \langle v_1, v_2, \ldots, v_n \rangle$, the strip of V consists of the $n-2$ triangles $\langle v_i, v_{i+1}, v_{i+2} \rangle$, for $1 \leq i \leq n-2$. See Figure 1.

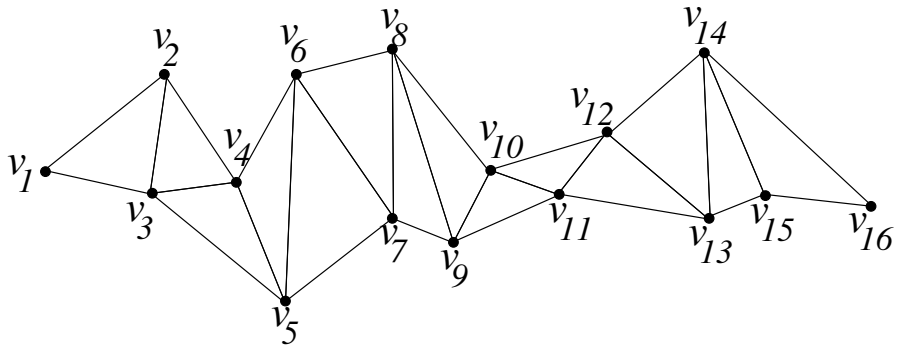

Figure 1: An x-monotone strip polygon.

Assume that you are given an n-sided, x-monotone simple polygon P as a sequence of vertices in counterclockwise order.

1. Give an algorithm to decide if P can be triangulated as a strip of the form given above.
2. Give a proof of correctness and derive the running time of your algorithm.

Note: The x-monotonicity will play a significant role in your algorithm and its proof of correctness. There is a simple linear time algorithm for this problem, but the proof of correctness is not completely straightforward. Your algorithm need not run in linear time. However, a more efficient algorithm will receive more credit.

6. **Implementation Choice** Write a program to generate and count the number of triangulations of a convex n-gon. The count-answer should be the Catalan number C_{n-2}. Your program should list the diagonals in each triangulation, but have an option to suppress the listing and just count (so that it can be checked for large n. Don't go crazy however, as the Catalan numbers are of exponential growth!). If you want to be specific, you can use the regular n-gon, but all convex n-gons have the same set of and number of triangulations. However you end up doing this is fine with me. As long as you list the diagonals in each triangulation, and list each triangulation exactly once, you have satisfied the task. As an example, your output for $n = 5$ could be (if you index the vertices $0, 1, 2, 3, 4$, see Figure 3.12 (a) in O'Rourke):

```
(0,2), (0,3)
(1,3), (1,4)
(2,4), (2,0)
(3,0), (3,1)
(4,1), (4,2)
```

Either way, just a list of the diagonal endpoint indices (not coordinates!) in some order (any order). No graphics is needed (but if you want to generate graphics to check yourself, that is of course fine).