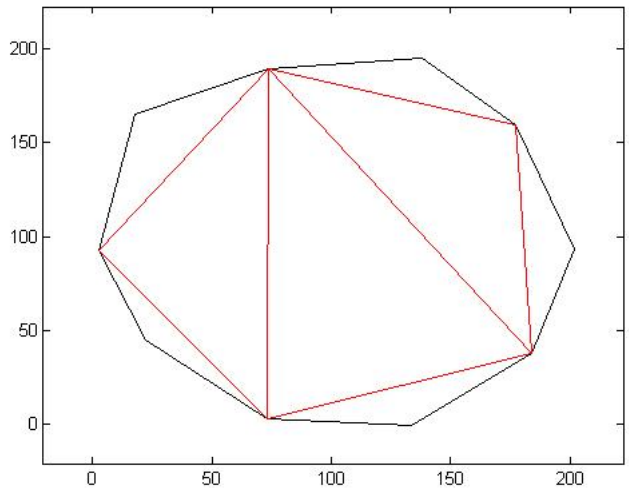
CS 405: Algorithm Analysis, Winter, 2013
Midterm, Part II
Take-home Exam, due March 4, 2013

The following figure depicts a 10-vertex convex polygon; the table lists the vertices proceeding counterclockwise from the rightmost vertex. The interior chords show a sample decomposition into triangles.

We consider only *convex* polygons in this assignment. A convex polygon has the property that the line between any two points, chosen from the interior or boundary of the polygon, lies entirely within the polygon. For a convex polygon, we define a *decomposition* as a selection of chords, each joining two non-adjacent vertices that divide the polygon into triangles and that do not intersect in the polygon interior. The challenge is to find a decomposition such that the sum of the triangle perimeters is minimum. We will call such a decomposition a *minimal decomposition*.



Vertex	x	y
0	202.1177	93.5606
1	177.3577	159.5286
2	138.2164	194.8717
3	73.9028	189.3758
4	17.8465	165.4303
5	2.4919	92.5714
6	21.9581	45.3453
7	72.9884	3.1700
8	133.3893	-0.3667
9	184.0190	38.2951

1. Give an argument showing that the number of candidate decompositions is exponential in n, the number of polygon vertices.

2. Find and explain a dynamic programming algorithm that finds a minimal decomposition in $\Theta(n^3)$ time or less.

3. Program your dynamic programming algorithm in your language of choice, being careful to use only those language constructs that require $\Theta(1)$ time. Your input/output should follow the example below.

   Example:

   Vertices are identified by their position in the input list, starting with vertex 0, proceeding counterclockwise around the polygon. Chords are identified as a vertex pair (a, b) with $a < b$. Letting v denote the vertex array shown in the figure above (available from the course web page as `polygon1.txt`), my program finds the minimal decomposition as

```
>> decompose(v);
Minimal sum of triangle perimeters =  2528.5090
7 chords are:
    1    9
    1    3
    3    9
    3    7
    7    9
    3    5
    5    7
Check: twice sum(chords) + poly perimeter =  2528.5090,
```

You can use `polygon1.txt` to check your program's correctness, but your submitted output should also show the output using `polygon2.txt`, which is also available on the course web page.

As a further check, I wrote another program that chooses random chords, subject to the non-intersecting property, to obtain the following competitors in polygon1.txt

```
>> random_decomposition(v, 10);
random  1:    2906.7288
random  2:    2752.6708
random  3:    2672.4656
random  4:    2736.4366
random  5:    2631.7518
random  6:    2625.2465
random  7:    2625.5235
random  8:    2947.4763
random  9:    2643.1255
random 10:    2681.6809.
```

Note that each perimeter exceeds the minimum found with the dynamic programming algorithm.