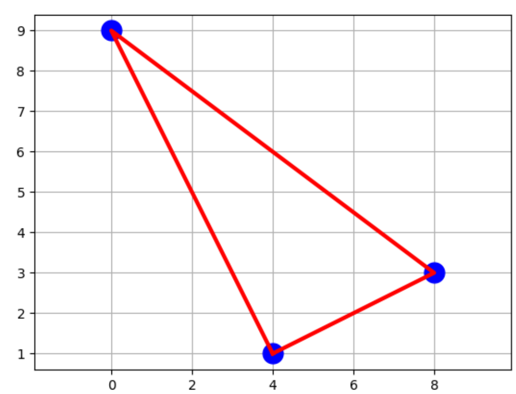
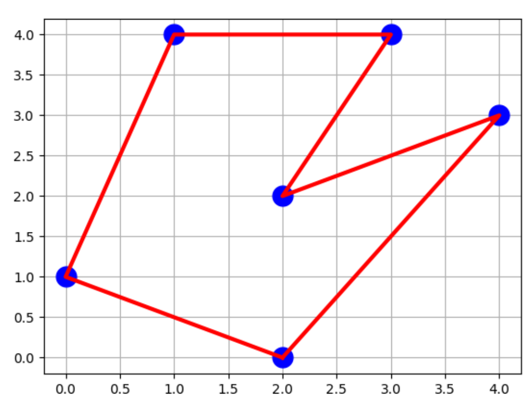
Assignment #10
Math 1800: Mathematical Programming in Python

Instructions: Choose 3 of the following problems to work on. Submit your responses as Python text files, with the extension `.py`. Each file should include your name and the problem number. This problem set is due Thursday, April 06, by midnight.

Several of the geometry problems refer to the *example triangle*. This is the triangle, which we discussed in class, whose vertices $((A, B, C)$ are $(4, 1), (8, 3), (0, 9)$, and whose area is 20. This object is depicted below.



Several of the geometry problems refer to the *polygon graph*. This object is depicted below.



- *Problem 10.0:* A random point p on the unit circle can be defined by getting a random number r and setting
$$p = (x, y) = (\cos(2\pi r), \sin(2\pi r))$$
Picking three such points will define a random triangle. Write a Python program which generates 100 random triangles this way, and prints out the mean and standard deviation of the set of 100 areas.
- *Problem 10.1:* An obtuse triangle contains an angle that is greater than 90°. Write a Python program which generates 100 random triangles whose vertices are on the unit circle. Print the number of these triangles which are obtuse.

- *Problem 10.2:* Write a Python program which creates a plot of the *polygon graph*.
- *Problem 10.3:* Divide the *polygon graph* into triangles, describing each by its vertex coordinates. Write a Python program which computes the area of each triangle, sums them up, and reports the area of the polygon.
- *Problem 10.4:* If a polygon of area A is divided up into n triangles, each of area a_i, each with centroid $c_i = (x_i, y_i)$, then the centroid of the polygon can be computed as $C = \frac{1}{A} \sum a_i \, c_i$. Divide the *polygon graph* into triangles, compute their areas, and determine the area A and centroid C.
- *Problem 10.5:* Estimate the area of the *example triangle* by random sampling. Surround the triangle by the square of dimensions $0 <= x <= 8$, $1 \leq y \leq 9$. This square has area 64. Compute $n = 1000$ random points (x, y), each of which is inside the square. Let k be the number of these points that are contained inside the triangle. Print your area estimate as Area $\approx \frac{k}{n} 64$.
- *Problem 10.6:* Use the Monte Carlo method to estimate the integral of $f(x, y) = 3x^3 + xy$ over the *example triangle.*
- *Problem 10.7:*. In class, a quadrature rule was discussed which uses six points. Use this rule to estimate the integral of $f(x, y) = 3x^3 + xy$ over the *example triangle.*
- *Problem 10.8:* In the first geometry discussion, we considered how to determine the distance from a point q to a line segment, and created a function `line_segment_distance(p0,p1,q)` to compute this. Suppose we wanted to compute the distance from a point q to a triangle. Write a Python function to determine this quantity. You need to use `line_segment_distance(p0,p1,q)` to measure the distance to each of the three triangle sides first, and then take the minimum. For your code, use the example triangle, and compute the distance from the points $q = (2, 5), (10, 3)$, and $(4, 5)$.
- *Problem 10.9:* Suppose line ℓ_0 is defined by points p_0 and p_1, and line ℓ_1 by points q_0 and q_1. Write a Python program which determines the equations that define these lines, and solve for their intersection point. The two equations have the form

  $$a_{0,0}x + a_{0,1}y = b_0 \quad \text{from line } \ell_0$$
  $$a_{1,0}x + a_{1,1}y = b_1 \quad \text{from line } \ell_1$$

  Your program should compute these coefficients from the point data, and then call the appropriate function to solve for (x, y). Then print this intersection point. Assume your data is $p_0 = (2, 2), p_1 = (6, 0), q_0 = (2, -1), q_1 = (6, 3)$, which means that the intersection point you are looking for is $(4, 1)$.