MCS 320 Project 3

Problem One. Write a Sage function problem1(L, t, n) which takes a list L of three vertices (coordinate points (x,y)) in a plane as a template for triangle; a tuple t representing a (x,y) coordinate point; and an integer n, specifying a number of iterations. See input format below. Implement the following steps in your function problem1(L, t, n):

```
0. Initialize an empty list P
1. Take a list of three vertices listed in L (coordinate points (x,y) ) to
   form a triangle, e.g. [(0,0),(10,0),(5,10)]
2. Select the point t as your initial point and consider that
   your current position.
3. Randomly select any one of the three vertex points in L by calling a random number
   generator simulating an ordinary six-sided dice:
       if 1 or 2 comes up, select the first point in L (e.g. [0,0])
       if 3 or 4 comes up, select the second point in L (e.g. [10,0])
       if 5 or 6 comes up, select the third point in L (e.g. [5,10]).
4. Move half the distance from your current position to the selected vertex chosen from
   problem 3 and make that your current position.
5. Store the vertex representing your current position in the list P.
6. Repeat from step 3 a 10000 times.
7. Once the process is complete, plot the points in the list P as a 2D plot.

NOTE: Your code should end with the following commands, which will produce the image,
      generated by the above described process.

      Pt = point(P, color='red', size=5)
      plot(Pt).show()
```

Include the answer (plot) for problem1([(0,0),(10,0),(5,10)], (4,2), 10000) as part of your answer.

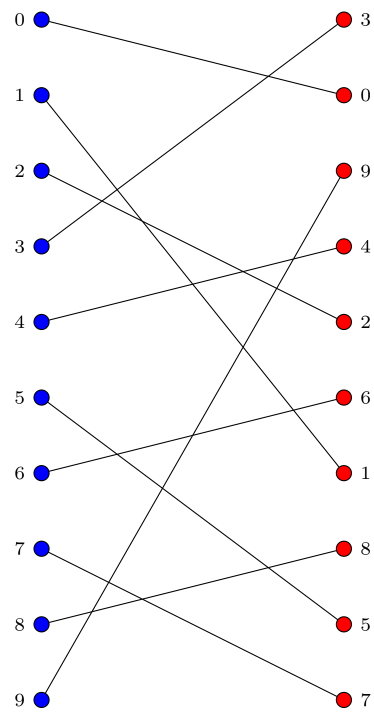
Problem Two. Write a Sage function problem2(f, g, h), which takes on input symbolic expressions in 4 variables x, y, z, w. The symbolic expression f represents a function. The symbolic expressions g and h represent constraints. The function problem2(f, g, h) finds and returns the maximum value of f, subject to the constraints g = 0 and h = 0. As your solution to this problem, provide the Sage function and the output for

```
problem2(4*x+4*y+9*z-2*w, 2*x+2*y+z+w==0, x^2+y^2+z-4==0)
problem2(x-y+2*z+2*w, x+y+z+w==0, x^2+y^2+z^2-3==0)
```

Hint: Recall the method of Lagrange Multipliers.

Problem Three. Consider the picture below. The *blue* points are arranged from least to greatest. The labels of the *red* points are exactly the same as the *blue* points. In particular, the red points are the blue points rearranged.

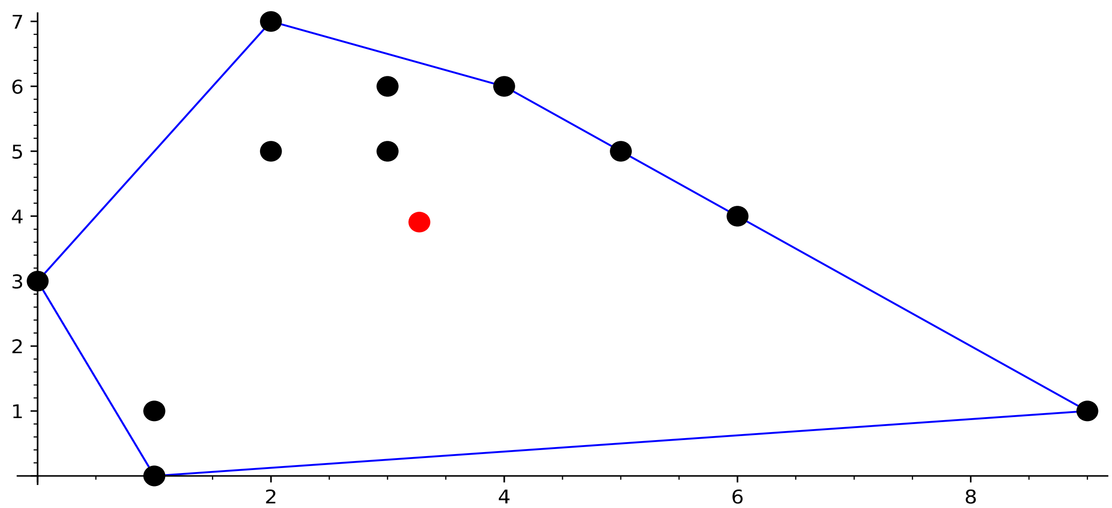
Write a function problem3(L), which takes a list of values L which corresponds to the order of the red points and counts the number of intersections of the line segments, between the blue and red points. In this illustration, we have 17 line crossings. As a result, problem3([3,0,9,4,2,6,1,8,5,7]) would return 17.



As your solution to this problem, provide the Sage function and the output for problem3([10,3,13,19,0,11,9,20,4,16,2,12,6,1,14,15,8,5,17,18,7]).

Problem Four. Consider a symbolic expression f in two variables $x$and y. Write a function problem4(f), which takes f on input. The function then coerces f into a polynomial ring (R) in variables x, y, over the field of rational numbers $\mathbb{Q}$. Once f has been coerced into the polynomial ring, Sage function exponents() returns a list of all exponents of the monomials in f as a list of coordinate tuples. Considering the monomial exponents as coordinate points in 2D, complete the function problem4(f) to compute the center of all coordinates (coordinate mean values) and returns the coordinate points and distances of the farthest point to the coordinate center and the closest.

As an illustration, consider the following picture, where $f = -3x^9y + x^6y^4 + x^5y^5 + 5x^4y^6 - x^3y^6 + 6x^2y^7 - 8x^3y^5 - 5x^2y^5 - 4y^3 - 2xy - 2x$, the coordinate center is the red disc, and the farthest and closes coordinate points to the center, along with their distance are ((9, 1), 6.4237421188834585) ((3, 5), 1.1244833524411801).



As your solution to this problem, provide the Sage function code and the output for

```
problem4(-2*x^19*y - 2*x^14*y^6 - x^4*y^16 - 9*x^17*y^2 - x^15*y^4 + 7*x^10*y^9 + 7*x^6*y^13
- x^12*y^6 - 2*x^9*y^9 + 9*x^8*y^10 + 9*x^5*y^13 - x^15*y^2 - x^6*y^11 - 12*x^3*y^14 - 3*x^9*y^7
- 7*x^7*y^9 + x^5*y^11 + 2*x*y^14 + 5*x^13*y + x^10*y^3 + 2*x^6*y^7 + x^12 + x^11*y - x^3*y^8
- 3*x^3*y^6 - 2*x^6*y^2 - x^5*y^3 + x*y^6 - x*y^3)
```

Project Guidelines, Submission Details, and Plagiarism Warning

This project is due on **Friday, October 23, 2020 at 9 AM**. No late submissions will be accepted and no extensions will be given!

Your solution to this project must consist of a single, computer typed (not hand-written), PDF document, called **project3.pdf**, containing the **Sage code** **and the answers** for the four problems. For each problem, write a Sage function, titled *def problem1(), def problem2(), etc*, with the appropriate input arguments. Do not change the function names. Upload the file **project3.pdf** through Blackboard. We must be able to copy and paste your code in order to evaluate it!

This project must be solved **individually**. Under no circumstances are you allowed to copy or to collaborate with anyone else. All submitted files will be automatically checked for plagiarism. Regardless of who copied from whom, all caught in the act of plagiarism will be penalized.

In particular, using internet resources of any kind is **not** allowed. Internet sites are routinely checked for similarity to your submission, both for code content and code logic. Changing code order or variable names will not prevent plagiarism detection. In addition, do not post any content of this project to any internet sites or make it public in any other form. **The content of this project is not in the public domain!**

You are free, however, to use our course resources, such as lecture notes, text books, and official Sage documentation during the solving of this project.

If you have questions about this project, come to my online office hours using the usual Blackboard link.