MCS 320 Project 2

Problem One. Write a Sage function problem1(f) which takes a polynomial $f(x)$ as a symbolic expression on input and prints the number of roots that are rational, irrational, real, and complex. Specifically, print format should be

```
Rational : number of rational roots as an integer
Irrational: number of irrational roots as an integer
Real : number of real roots as an integer
Complex : number of complex roots as an integer
```

Include the answer for the polynomial $f(x) = x^{10}+x^9-7x^8-8x^7+6x^6+14x^5+22x^4+8x^3-7x^2-15x-15$.

Problem Two. Recall that for an integer n we say that $n \equiv a \mod k$ if when we divide n by k the remainder is a. Every prime number p other than 2 satisfies either $p \equiv 1 \mod 4$ or $p \equiv 3 \mod 4$. For instance, there are 80 prime numbers ≤ 1000 that are $\equiv 1 \mod 4$ and 87 primes ≤ 1000 that are $\equiv 3 \mod 4$. Further, there are 609 prime numbers ≤ 10000 that are $\equiv 1 \mod 4$ and 619 that are $\equiv 3 \mod 4$. While it seems like the number of primes $\leq n$ that are $\equiv 3 \mod 4$ is always larger than the number that are $\equiv 1 \mod 4$, the race between the two switches back and forth infinitely often. The smallest n for which there are more primes $\leq n$ that are $\equiv 1 \mod 4$ than there are that are $\equiv 3 \mod 4$ is $n = 26861$.

For this problem, write a Sage function problem2(k,a,b) taking in values (k, a, b) which returns the smallest number n so that the number of primes $\leq n$ that are $\equiv a \mod k$ is greater than the number of primes $\leq n$ that are $\equiv b \mod k$. For an example, the previous paragraph tells us that we have

```
Input: problem2(4,1,3)
Output: 26861
```

For a smaller example, we also have problem2(5,1,4) outputs 11.

As your solution to this problem, provide the Sage function and the output for problem2(5,1,2).

Problem Three. Let R denote the ring of polynomials in the variable x with coefficients in the rational numbers $\mathbb{Q}$.

(a) Write Sage program random_poly(n) which takes in a positive integer n and outputs a random polynomial of degree n whose coefficients are independently $-1, 0$ or 1 with equal probability; this output should be an element of R. For instance, one possible output is given below:

```
Input: random_poly(4)
Output: x^4 - x^3 + x -1.
```

Show an output for random_poly(5).

(b) Write a Sage program problem3(n,M) which takes in integers (n, M). Your program should generate M random polynomials of degree n using random_poly(n) and count how many of them are irreducible. Do not print or output the polynomials themselves, only return the final count. An example output is given below:

```
Input: problem3(11,1000)
Output: 430
```

As your solution to this problem, provide the Sage function and the output for problem3(15,1000).

Problem Four. Recall that

$$\frac{\pi^2}{6} = \sum_{k=1}^{\infty} \frac{1}{k^2} \,.$$

Write a function problem4(m) that takes in a number m and outputs a numerical approximation to the sum

$$\sum_{k=1}^{m} \frac{1}{k^2} = \frac{1}{1} + \frac{1}{4} + \ldots + \frac{1}{n^2} \,.$$

A sample input/output is given below:

```
Input: problem4(100)
Output: 1.63498390018489
```

As your solution to this problem, provide the Sage function and the output for problem4(10000) and use the function cputime() to find its runtime.

Project Guidelines and Submission Details

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

Your solution to this project must consist of a single PDF document, called **project1.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 **project2.pdf** through Blackboard. **No other format will be accepted**.

This project must be solved **individually** but you may brainstorm ideas with each other. Under no circumstances are you allowed to copy or to collaborate with anyone else beyond big-picture discussion. 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. Using the internet resources is also off limits. However, you are free 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 Zoom meeting.