General instructions: No problem on this homework is intended to take more than a half hour, and most should take under fifteen minutes. If they are taking longer than this, please come ask for help. If you are new to Python/Sage, do the Warm-up problems and the Regular problems. If you already know how to program, do the Regular problems and the Challenge problems.

For homework problems you can always use code that we have previously developed in class. Also, here are some new Sage commands that will be useful:

- \texttt{randint(a,b)} returns a random integer in the interval \([a, b]\)
- \texttt{is\_prime(a)} is a built-in routine that checks whether the integer \(a\) is prime. This has the same function as our \texttt{primeset}\texttt{est} code, but is much faster.
- \texttt{gcd(a,b)} returns the greatest common divisor of the integers \(a\) and \(b\).
- If \(l\) is a list then \texttt{len(l)} returns the length of the list.
- If \(l\) is a list, then \texttt{l\_count(a)} returns the number of times \(a\) appears in the list \(l\).
- If \(l\) is a list, then \texttt{l\_append(a)} appends \(a\) to the end of the list \(l\). This function does not actually return anything, it just changes \(l\).

Warm-up problems

1. Write a Sage function \texttt{num\_div} that counts the number of divisors of a given positive integer \(n\). \([\text{Check: } \texttt{num\_div}(10)=4]\) How many divisors does 10,000 have?

2. Write a Sage function \texttt{prim\_interval} that takes a positive integer \(n\) and determines if there is a prime number in the interval \([n, n+13]\). Is there a prime between 1025 and 1038?

3. Write a Sage function \texttt{int\_max} that takes two positive integers and outputs the maximum of the two.
4. Write a Sage function `list_max` that takes a list of numbers as input and returns the maximum value from the list. The following outline gives one way to do this; fill in the places that have question marks. (Note that the apostrophe used twice in line 3 is the one below the double quotes " on your keyboard).

```python
def list_max(l):
    if len(l)==0:
        print 'Error: no elements'
        return
    if len(l)==1:
        return ??????
    temp_max=l[0]
    i=??
    while i<=len(l)-1:
        b=l[i]
        if b>temp_max:
            temp_max=????
            ????
    return temp_max
```

Check your code by trying it in several examples. (Note: The code I wrote is intentionally a little longer in places than it needs to be. Can you see how to simplify it?)

Regular problems

5. Recall that two integers $a$ and $b$ are said to be relatively prime if $\gcd(a, b) = 1$. If we pick random integers between 1 and $N$, what is the probability that they are relatively prime? Write some Sage code that will help you fill in the following table (I suggest doing between 100,000 and 1,000,000 trials for each):

<table>
<thead>
<tr>
<th>$N$</th>
<th>100</th>
<th>1000</th>
<th>$10^4$</th>
<th>$10^5$</th>
<th>$10^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>probability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you follow the method from class, you will need an `experiment` function and the `trials` function. [Check: As $N$ gets very large, the probability should approach $\frac{6}{\pi^2}$].

6. For a randomly chosen interval of 14 consecutive integers under 1,000,000, what is the probability that the interval contains at least one prime? Write some Sage code that approximates this. [Check: The probability of finding a prime in an interval $[a, a + 5]$ is about 0.42].

7. What is the probability that if we flip a coin 10 times, at least seven heads occur? Write a Sage function that takes an integer $n$ and approximates this probability, by doing the 10-coin-flip experiment $n$ times. [Check: The probability that at least five heads occur is $\frac{319}{512}$].
Fill in the following table with approximate probabilities for the number of heads in a 10-coin-flip experiment:

<table>
<thead>
<tr>
<th># heads</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>probability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Challenge problems

8. If you have 23 people in a room, the probability that at least two have the same birthday is over 50%. Most people find this hard to believe, so let’s use Sage to investigate.

   (a) Write some Sage code that approximates this probability by performing the experiment $N$ times, where you get to decide $N$. A randomly chosen person’s birthday can be modeled as a random integer between 1 and 365 (let’s ignore leap years for this problem).

   (b) How many people do you need to have for the probability that at least two have the same birthday is at least 80%? Use Sage to find out.

9. A bug starts at the origin on the number line. He flips a coin to decide whether to move left or right one unit. He repeats this process $N$ times. Write Sage code to answer the following questions:

   (a) What is the approximate probability that after 20 steps the bug is more than 10 units away from the origin?

   (b) Theoretical arguments say that if the bug takes $X$ steps then the average distance he ends up from the origin is around $\sqrt{\frac{2}{\pi} \cdot \sqrt{X}}$, at least when $X$ is large. Write some Sage code that tries to confirm this. [Warning: Note that this problem concerns averages of outcomes, not probabilities].