Imperial College London, Department of Mathematics

Scientific Computing (M3SC)

Calum S. Skene

January 23, 2017

1 Question 1: exerTools.py

The purpose of this question is to complete functions in the supplied file exerTools.py. These functions will be used in the later more advanced questions.

1.1 RabinKarp

In its current state running the Rabin-Karp algorithm will print to the screen the locations in a string where the first character of a pattern matches a section of the string. For example running

```
RabinKarp('1234123','123')
```

will print 0 and 4 to the screen since ‘123’ matches a section of the string beginning at the zeroth and fourth location in the string ‘1234123’. The first exercise is to change the Rabin-Karp algorithm, contained in the file exerTools.py, to return the match locations as a list. You may find it aesthetically pleasing for further exercises to remove the print statements from this code. You may test your code by running

```
pytest test_RabinKarp.py
```

1.2 isPrime

Complete the isPrime function so that calling

```
Bool = isPrime(p)
```

will give Bool as False if p is not prime and True if p is prime (1 is not a prime). You may test your code by running

```
pytest test_isPrime.py
```

1.3 primeList

Using the isPrime function complete primeList so that

```
List = primeList(N)
```

will give List as a list of the number of primes up to and including N. You may test your code by running

```
pytest test_primeList.py
```

2 Question 2: Primes in π

Open the exer2.py file. Generate a list of the primes up to 10000 using primeList. Find out how many times each of the primes in this list occur in the first 100000 digits of π. Plot the number of times each prime occurs against that prime in a log-log (base 10) plot using matplotlib. Can you explain the resulting plot?

3 Question 3: Birthdays in π

Open the exer3.py file and search for your Birthday in the first 100000 digits of π by writing it in a four digit form formed by concatenating the day and month using leading zeros if necessary. For example if you are born on the first of January you would search for the pattern 0101.
Search for all possible Birthdays. What Birthday(s) occurs the most number of times?

4 Bonus Question: Scaling

Open the exer4.py file. Calculate the time the algorithm takes to find the supplied four patterns (labelled p1,p2,p3 and p4) in the genetic material string and plot the time against the length of the pattern using matplotlib. Repeat this calculation using the naiveSearch function and plot this on the same graph. Does this agree with the theoretical scaling results shown in lectures?

Calculate the above graph again using a smaller prime (e.g 3) for the modulus of the hash function. What is the effect and why?