Final Quiz - C1010

Time - 6 hours

Answer all questions

1. Define classes named Ellipse and Rectangle which can be constructed using a pair of length measures. Both classes should have methods to compute area and perimeter. For the class Ellipse, for defining a method for perimeter computation, use the formula: $p = \sqrt{2}\,\pi\,\sqrt{a^2+b^2}$ where a and b are lengths of semi-major and semi-minor axis, respectively.

2. For a given number, check whether the number is a prime number or not. When the number is not prime, find all the factors of the number.

3. Cramer's rule is a method of computing the determinant of a matrix. Consider an $n \times n$ square matrix M. Let M_{ij} denote the element in the i-th row and j-th column of M, and let C_{ij} be the cofactor of M by removing the i-th row and j-th column from M and using appropriate sign.

The cofactor C_{ij} is defined as

$$C_{ij} = (-1)^{i+j} \det(m_{ij})$$

where m_{ij} is the minor formed by removing i^{th} row and j^{th} column of M. Then $\det(M)$ is determined by the expansion:

$$\det(M) = \sum_{j=1}^{n} M_{ij} C_{ij} = M_{i1}C_{i1} + M_{i2}C_{i2} + \cdots + M_{in}C_{in}$$

Write a function my_rec_det(M), where the output is $\det(M)$ using recursion. The function should use Cramer's rule to compute the determinant, not Numpy's function.

4. Open the site: http://towersofhanoi.info/Animate.aspx

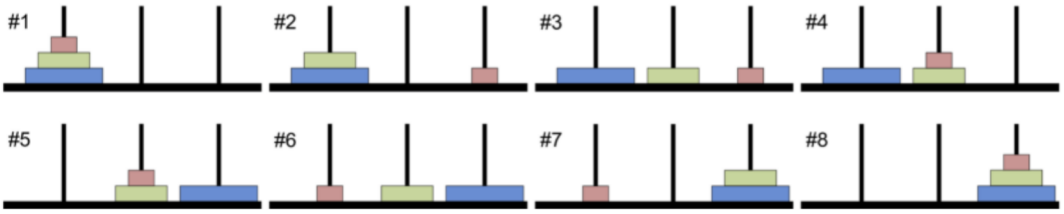
This is the well-known Towers of Hanoi problem that allows multiple pegs and disks. In the simplest form, Towers of Hanoi problem consists of three vertical rods, or towers, and N disks of different sizes, each with a hole in the center so that the rod can slide through it. The disks are originally stacked on one of the towers in order of descending size (i.e., the largest disc is on the bottom). The goal of the problem is to move all the disks to a different rod while complying with the following three rules:

a. Only one disk can be moved at a time.

b. Only the disk at the top of a stack may be moved.

c. A disk may not be placed on top of a smaller disk.

See the solution with three pegs and three disks. For N=3 pegs, there is a solution provided in the "Python Numerical Methods" book in section Recursive Functions: Divide and Conquer.



Write a program for N=4 pegs (towers) and number of disks = 4.

5. Given an array of integers, write a function that returns TRUE if there is a triplet (a,b,c) that satisfies $a^2 + b^2 = c^2$.

6. Consider a circle with unit radius ($r = 1$). Use numpy.random.rand function to generate coordinates (x, y). Check if $x^2 + y^2$ lies within the circle. Run this again and again for some large user-specified number of iterations. Keep track of the number of times (x, y) lies within the circle. If total number of iterations is N and total number of times (x, y) are within circle is n_{inside}, then evaluate $(4\,n_{\text{inside}})/N$. Is this close to a special mathematical constant?