Programming Assignment #3

Mathematical and Probabilistic Algorithms

For this assignment, you are to write 5 small programs:

1. **Computing Π (Pi) probablistically**

   Think of a square, with sides = 2, centered at the origin with a circle inscribed within the square. If you throw N darts that land in the square, some of them, M, will land inside of the circle. Since the area of the circle is Πr^2 (which is Π since r = 1) and the area of the square is 4, the ratio of the circle's area to the square's area is Π/4. M/N should approximate Π/4, so 4M/N should approximate Π.

   Simulate the throwing of the darts and printout the approximate value of Π computed for various iterations such as 1000, 10,000, 100,000, 1,000,000 and 100,000,000 (if time allows (it should)).

2. **Testing for Prime Numbers**

   You can test if a number P is a prime by checking to make sure P % n (n is an integer) is not zero for lots of values 1 < n < P. Write a program to input a number (or generate a number randomly) and test it against k random values. Pick a bunch of known composite numbers and see how accurately your program rejects them. See if the results improve for larger values of k. Try k = 10, 100, 1000 and 10,000. You can easily try the program against large composite numbers by including:
   1. Even numbers
   2. Numbers whose final digit is a 5
   3. Numbers where the sum of the digits is divisible by 3.
   4. Number constructed as the product of two positive integers.
   5. Randomly generated numbers X that are then adjusted as follows:
      `X = X - X%K` <-- X will now be divisible by K.

3. **Searching an Array**

   Create an array of 1000 ints. Search for a value known to be in the array by generating an index at random and testing to see if the searched for int is in the array at that index. Write your program to make at most 5000 guesses. Try 100 different searches and compute the average number of comparisons the program has to do. You can easily pick a target by randomly selecting a value from the array.

4. **Monte Carlo Integration**

   For this problem, you are to determine the approximate integral of a function on an interval two different ways using probabilities.

   - **Dart throwing:** Construct a rectangular region around the function. Pick random points in the rectangle. Find the percentage of these that are in the area under the function. The area under the function (the function's integral on this region) can be found by multiplying this percentage by the rectangle's area.
   - **Mean of values at random locations:** Compute the mean value of the function at random locations in the interval and multiply this value by the interval's width.

   Compare these two methods to the trapezoid method. Compare both acccuracy and running times. Make sure to use interesting functions for your tests.

5. **8 queens problem**

   Solve the 8 queens problem by gluing k random queens to the board and placing the other 8-k queens using backtracking. What value for k gives the best result (on avereage)? How does the running time compare to the traditional backtracking algorithm?

What to hand in

Turn in a copy of your program[s] along with a write up of the results.

Due Date:

Email your program and turn in the paper in class by Thursday 2019.10.24.