Randomness and Computation

Problem 29

Ryan Burmeister

February 2, 2016

Python: Python random’s class implements the Mersenne Twister as its PRNG. The period of the of PRNG is $2^{19937-1}$, and the default seed is set by using the current system time which is done when the module is first imported. If a random source is available, then that source is used to supply seeds. Therefore, A ”true” entropy source is only used if provided be the OS.

Java: Java’s Java.util.Random implements a linear congruential formula with 48-bit seeds. The period is only $2^{4}8$. To generate a random number, the documentation states that it sets the seed value to one ”likely to be distince from any other invocation of this constructor.” After further digging, the code consists of a call to seedUniqueifier(). This constructor follows a method from ”Table of Linear Congruential Generaors of Different Sizes and Good Lattice Structure (1999, L’Ecuyer)” to generate its seed. The iniital value is set to 8682522807148012, and every subsequent value is computed by multiplying the current seed value by 1817783497276652981 and xor-ing that number with the current time in nanoseconds.