Programmatic Entropy:
Exploring the Most Prominent PRNGs

Jared Anderson, Evan Bause, Nick Pappas, Joshua Oberlin

$ 5HFDS RI 5DQGRPQHVV

Random number generating algorithms are rated by two primary measures: entropy - the measure of disorder in the numbers created and period - how long it takes before the PRNG begins to inevitably cycle its

number generation to not sacrifice performance of the product the PRNG is required for. However, in the real world PRNGs must also be evaluated for memory footprint, CPU requirements, and speed.

In this poster we will explore three of the major types of PRNGs, their history, their inner workings, and their uses.

7KH 0HUVHQH 7ZLVWHU

The Mersenne Twister is the most widely used general purpose pseudorandom number generator today. A Mersenne prime is a prime number that is one less than a power of two, and in the case of a mersenne twister, is its chosen period length (most commonly 2^{19937}–1). There are only 50 such numbers known to exist today, with the most recent, $2^{77,232,917}$-1, only being found on January 3rd of this year. Makoto Matsumoto and Takuji Nishimura developed the algorithm in 1997 to overcome the flaws found in older PRNGs, being the first PRNG to provide high entropy, long period random number generation in little time. Because of this, the Mersenne Twister is the default pseudorandom number generator for software systems such as Microsoft Excel, GAUSS, GNU, IDL, MATLAB, Python, R, and Ruby.

There are many reasons to choose the mersenne twister over other PRNGs. As stated, it has a truly impressive speed and quality combination, producing even 64-bit floats 20x faster than hardware based solutions, while also passing statistical tests for randomness like TestU01 and Diehard. It is also patent-free, so it and variants of its base algorithm can be used freely without worry of cost or expensive hardware.

The Mersenne Twister is, however, not a silver bullet for all PRNG needs. While a variant, CryptMT, exists, it is normally not cryptographically secure, disallowing its use where security is a major concern such as password encryption or gambling. It also requires a large state buffer of 2.5 KiB, and has mediocre throughput by modern standards, meaning it shouldn't be used on hardware with too little buffer or in situations where large streams of random numbers are needed.

Mersenne Twister takes a seed which is initialized into a 624 x 32bit

Linear congruential generators form a series of numbers using the