W wynikach skupię się głównie na tym co się „nie udało” (w ramach zwięzłości). Zostawiam również krótką notkę, za co poszczególny test odpowiada (żeby było wiadomo co się udało).

Zarówno dla Mersenne Twistera jak i LCG przeprowadziłem 5 testów (na podanej w poleceniu stronie), każdy z 1 000 000 bitów. Dla SHA-1 test przeprowadziłem jedynie raz, z uwagi na to, że dla jednego nazwiska hash będzie ten sam => nie ma fizycznej możliwości na otrzymanie innego wyniku używając tej samej strony i tego samego nazwiska.

Za co odpowiadają poszczególne testy NIST?(w DUŻYM uproszczeniu)

1. Sprawdza czy ilość 1 i 0 jest ”+/-” = 50%.
2. Sprawdza czy 1 i 0 są ”+/-” równomiernie rozłożone tj. nie ma sytuacji, gdzie wszystkie jedynki są w pierwszej połowie ciągu znaków, a wszystkie zera w drugiej.
3. Sprawdza czy ilość ciągów 0 i 1 różnej długości (np. #0111110 itp.) jest odpowiednia.
4. Sprawdza czy częstość z jaką pojawiają się najdłuższe przewidywane ciągi 1.
5. Sprawdza czy istnieją powtarzające się wzory (”patterns”).
6. Sprawdza czy powyższe powtarzające się wzory są blisko siebie.
7. Sprawdza czy istnieją konkretne (ustalone) ciągi bitów i czy ich liczba jest odpowiednia (ciągi te nie mogą na siebie nachodzić).
8. Sprawdza czy istnieją konkretne (ustalone) ciągi bitów i czy ich liczba jest odpowiednia (ciągi te mogą na siebie nachodzić).
9. Sprawdza odległości między takimi samymi wzorami.
10. Sprawdza długość ” a linear feedback shift register (LFSR)” potrzebnego do wygenerowania danego ciągu bitów.
11. Sprawdza częstość występowania wszystkich możliwych nachodzących na siebie m-bitowych wzorów. Dla każdego takiego wzoru, częstość występowania powinna być taka sama jak innych wzorów tej samej długości. Liczba takich wzorów powinna być odpowiednia.
12. Sprawdza entropie (czyli także ”randomness”). Aproksymuje przez podzielenie # ww. m-bitowych wzorów przez # m+1-bitowych wzorów. Wynik ten porównuje się z oczekiwanym (maksymalnie ~ln2).
13. Sprawdza czy 1 lub 0 występują w dużych ilościach na początkowych i/lub końcowych etapach ciągu lub czy są równomiernie wymieszane w całym ciągu.
14. Sprawdza ile razy w danym cyklu występuje dana ”cumulative sum”.

Dokładniejsze informacje/ew. niedopowiedzenia można znaleźć/rozwiać tutaj: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf>

MERSENNE TWISTER

Generator bez większych problemów przechodzi (prawie) wszystkie testy.

Dwa razy generator nie przeszedł testu 3, lecz P-value jest tam większa od 1, co jest ciężkie do zinterpretowania i (myślę że) skłaniałoby raczej do spodziewania się błędu po stronie strony internetowej (implementacji testu NIST), aniżeli samej metodologii czy wartości wyrzuconych przez generator.

Raz nie przeszedł testu 4, lecz był bardzo blisko (zabrakło 0,002 w P-value).

Raz nie przeszedł testu 8, lecz był bardzo blisko (zabrakło 0,002 w P-value). Test sprawdza jak bardzo da się skompresować dany ciąg bitów, a Mersenne Twister znany jest z „odporności” na kompresowanie. Nie uznałbym tego negatywnego wyniku za „bardzo” istotny.

W jednej próbie wystąpił błąd w teście 13 i 14 (ciężko mi powiedzieć dlaczego, ale spodziewam się winy po stronie strony).

Raz „oblał” test 14 (zabrakło 0,005). Oznacza to, że w tym teście była lekka tendencja do przebywania w danym „stanie” ciut częściej lub rzadziej niż zakładano od PRNG.

Wnioski z tych testów płyną następujące: Mersenne Twister jest bardzo dobrym generator liczb pseudo losowych, czasami zdarzy mu się wypaść poza „strefę przyzwoitości” i nie przejść danego testu. Jednak wypadki takie nie są ani częste, ani bardzo drastyczne (P-value są bardzo niewiele poniżej zakładanego zakresu).

LCG

Generator przechodzi większość testów.

Raz nie przeszedł testu 3 (P-value >1 => możliwa wina po stronie strony).

Raz nie przeszedł testu 11, P-value było bardzo niskie. Z definicji LCG jest dość podatny na wpadanie w bardzo krótkie cykle, pętle; stosunkowo łatwo aby występowała duża powtarzalność. Możliwym jest, że coś takiego miało właśnie miejsce i stąd bardzo niska entropia (choć dziwi pomyślne przejście innych testów).

Raz nie przeszedł testu 13 (zabrakło 0,004). Jednak przeszedł test 14. Testy te są w pewien sposób „bliźniacze”, 13 bada 8 stanów (stany powyżej najwyższego lub poniżej najniższego są liczone jak najniższy lub najwyższy) a 14 aż 18 stanów. Możliwym jest więc, LCG „lubi wpadać” w stany wyższe i niższe stosunkowo często w porównaniu do tych bliskich zeru.

Wnioski: LCG jest dość dobrym generatorem liczb pseudo losowych, jednak nie tak dobrym jak Mersenne Twister(MT). Rozkład jego „stanów (testy 13 i 14)” nie jest tak dobry jak dla MT (wysoka ilość błędów dla tych testów również nie napawa optymizmem). Natomiast największą jego wadą jest łatwość wpadania w krótkie („mało losowe”) cykle.

SHA-1

Zacznę od tego, że nie jest on wstanie przejść wielkości testów ze względu na potrzebną ilość bitów do ich przeprowadzania (brakuje kilku rzędów wielkości generowanego hashu).

Nie przejściem testu 3 (P-value powyżej 1, znów obarczyłbym stronę).

Nie przejście testu 4 oznacza, że nie pojawił się odpowiednio długi ciąg 1 (co może występować często, jeśli nasz hash ma zależeć od nazwiska).

Test 11 wskazał niską entropię => stosunkowo dużą „powtarzalność” możliwych wzorów.

Wnioski: O ile hashowanie ma swoje niewątpliwe plusy i zastosowania, o tyle nie uznałbym go za dobry sposób generowania liczb losowych.