1 Random Number Generators (RNG or PRNG)

1.1 Truly random number generators

- Slow
- Not repeatable (without storing all of the numbers)

1.2 Middle square generators

First generator. Von Neumann: “Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin.”

Middle square generator
\{64 bit arithmetic to produce 32-bit random numbers.\}
\begin{verbatim}
 r ← seed \{Initial 32-bit value\}
 function rand(32-bit value r)
     r ← middle 32 bits of \(r^2\)
     return(r)
 end function
\end{verbatim}

Middle square Weyl sequence generator
\{64 bit arithmetic to produce 32-bit random numbers.\}
\begin{verbatim}
 w ← 0 \{64 bits\}
 s ← ‘64-bit irrational number’
 r ← seed \{Initial 32-bit value\}
 function rand(32-bit value r)
     w ← w + s
     r ← middle 32 bits of \(r^2 + w\)
     return(r)
 end function
\end{verbatim}

1.3 Linear Congruential Generator (LCG)

Linear Congruential Generator (LCG)
a, c, m are (carefully chosen) constants.
r ← seed \{Initial value\}
function rand(r)
    r ← ar + c mod m
    return(r)
end function

IBM “truly horrible” RNG, RANDU: \(a = 2^{16} + 3, c = 0, m = 2^{31}\)
2 Random permutations

Fisher-Yates shuffle (aka Knuth Shuffle)

\[
\text{for } i = n \text{ downto } 2 \text{ do } \\
\quad j \leftarrow \text{random}(1,i) \\
\quad A[i] \leftrightarrow A[j] \\
\text{end for}
\]

Wrong but common shuffle

\[
\text{for } i = 1 \text{ to } n \text{ do } \\
\quad j \leftarrow \text{random}(1,n) \\
\quad A[i] \leftrightarrow A[j] \\
\text{end for}
\]