

## アルゴリズムとデータ構造 擬似乱数発生法

- 乱数とは何か

サイコロを次々に投げて得られるようなでたらめな数の列

例 1: 6,2,9,5,1,4,1,3 . 例 2: 1,1,1,1,1,1,1,1 例 3: 3,1,4,1,5,9,2,6,5,3,5,...

- 乱数の使用目的

現実現象を模倣するシミュレーション  $\implies$  現象を数式で記述 . それを計算機で解く .

巨大な母集団からランダムにサンプリングをおこなう .

コンピュータゲーム : 敵をでたらめに動かす  $\implies$  でたらめな数字の列が必要

- 物理乱数と擬似乱数 (pseudo random number)

1. 物理乱数 : 現実の世界で , でたらめに見える現象を利用して乱数を作る . 雑音 , 気温 .

2. 擬似乱数 : コンピュータはあらかじめ決められたプログラムによってしか動かない

$\implies$  乱数のように見える数を作り出す .

- 擬似乱数の生成アルゴリズム

平方採中法 (middle-square method) . John von Neumann が開発 (1946) .

線形合同法 (linear congruential generator) . Derrick H. Lehmer 教授が 1960 年頃に開発 .

$$x[i+1] = ( a*x[i] + c ) \% m$$

ここで  $\% m$  は  $m$  で割った余り (mod とも書く) .

例 :  $a = 12, c = 0, m = 31$  . 初期値  $x[0]=1$  (乱数の種)

$$x[1] = ( 12*x[0] ) \% 31 = 12 \% 31 = 12$$

$$x[2] = ( 12*x[1] ) \% 31 = 144 \% 31 = 20$$

$$x[3] = ( 12*x[2] ) \% 31 = 240 \% 31 = 23$$

$$x[4] = ( 12*x[3] ) \% 31 = 276 \% 31 = 28$$

$$x[5] = ( 12*x[4] ) \% 31 = 336 \% 31 = 26$$

$$x[6] = ( 12*x[5] ) \% 31 = 312 \% 31 = 2$$

となり , 1, 12, 20, 23, 28, 26, 2... と乱数のように見える数列が得られる .

標準擬似乱数 rand() :  $a = 1103515345, c = 12345, m = 2^{31} = 2147483648$  .

- 乱数が満たすべき性質 : 等確率性 , 統計的独立性 (以前に出現した数値に依存しない) .

- 望ましい乱数生成法 : 高速 (計算が容易) , 再現が容易

- メルセンヌ・ツイスター (Mersenne Twister) 法 . M .Matsumoto & T. Nishimura