'Random' Thoughts

Date
DELTA Pg No

What is Randomness?

"'2' is a random no." – What do you mean by that?

1, 2, 4, 8, 16 looks like its not random (powers of 2)

1, 2, 4, 8, 16, 31, 57, 99, 163 looks like is random, but is it?

	1	2	4	8	16	31	57	99	163
difference →	1	2	4	8	15	26	42	64	
Repeat →	1	2	4	7	11	16	22		
repeat →	1	2	3	4	5	6			
Repeat →	1	1	1	1	1				

So, it's not random!!

How do you know that a sequence is actually random or you ~~are~~ have not found any pattern yet?

Do you know any function in math that gives out (random numbers)?
↳ what do they even mean?

How can we proof that there ~~is~~ can never be a pattern in a given sequence? maybe you haven't discovered it yet!

Don't go to the next page, you might get satisfied by my answers, even though I am not satisfied by them. Try to think, and share it!

Random → unpredictable (prediction space is large or prediction takes a lot (like a lot) of time)

optimisation : time - space - accuracy

So, instead of predicting accurately whether a coin toss will result in a head or a tail (by applying laws of rotation, friction etc) before the coin falls, we can approximate the outcome and save a lot of time. So, then we can say that outcome of a fair coin toss is "unpredictable" as each outcome is equally likely.

So, we can use coin toss to generate randoms nos. (binary numbers).

Heads → 0
Tails → 1

Random sequence ⇒ H T T H T H H H T T H H T H
Random binary no ⇒ 0 1 1 0 1 0 0 0 1 1 0 0 1 0
Random decimal no ⇒ $2^{12} + 2^{11} + 2^9 + 32 + 16 + 2$
$= 6706$

But, how do code a coin toss?

→ use import random

Date
DELTA Pg No.

After understanding that we don't really understand random numbers, maybe we should also question how 'import random' works?
(Mersenne Twister)

There are a few ways to generate pseudo random numbers, like linear congruential generator (lcg) (once you know it, you will know why it's called that)

Working :- we give a number ~~in the~~ to begin and then we make a sequence (pseudo random) using it such that $X_{n+1} = (aX_n + b) \bmod m$
here a, b and m are fixed and X_0 is given

So, if $a = 6$
$b = 7$
$m = 11$
$X_0 = 2$

gives remainder

$X_1 = (12+7) \bmod 11 = 19 \bmod 11$
$X_1 = 8$
$X_2 = 48 + 7 \bmod 11 = 55 \bmod 11$
$X_2 = 0$
$X_3 = 7 \bmod 11 = 7$
$X_3 = 7$
$X_4 = (42+7) \bmod 11 = 49 \bmod 11$
$X_4 = 5$
$X_5 = (30+7) \bmod 11 = 37 \bmod 11$
$X_5 = 4$
$X_6 = 31 \bmod 11 = 9$
$X_7 = 61 \bmod 11 = 9$
$X_8 = 61 \bmod 11 = 9$
$X_9 = 61 \bmod 11 = 9$

} Not random anymore!

Play with different values for 'a', 'b', 'm' and 'X_0' to see any patterns.

Math

1.) $0 < a < m$
2.) $0 < b < m$
3.) $0 < X_0 < m$

1.) if $a > m$

$a = m + x$

$$((m+x)X_0 + b) \bmod m$$
$$= (mX_0 + xX_0 + b) \bmod m$$
$$= 0 + (xX_0 + b) \bmod m$$

and $0 < x < m$, you can take $a > m$ but (mod m) will anyways treat it as some no. b/w 0 and m.

2.) if $b > m$

<same>

3.) if $X_0 > m$

<same>

It is easy to observe that if 'm' is prime we get more unique numbers than when 'm' is not prime.

$(aX + b) \bmod m$

$= \underbrace{aX \bmod m}_{\text{the number before 'a' i.e. a-1 should be divisible by all prime factors of 'm' so that 'a' is not divisible by them and this part i.e. aX mod m doesnot become 0; and then we will get the same number all the time. (if m is not prime)}} + \underbrace{b \bmod m}_{\Downarrow}$

b, m should be co-prime if they are not, then we can just have aX mod m to generate random nos. & 'b' is just wasting our time. (if m is not prime)

a > m
t as

we can use current time as X_0 to have more randomness

Btw, $X_n = (aX_{n-1} + b) \bmod m$

$(aX_{n-1} + b)$ ⇓ linear equation (equ. of a line)

mod → congruential ($A \equiv B \pmod C$)

$\Rightarrow A \bmod C = B \bmod C$

$A \bmod B = \text{rem}(A, B)$