

## Re: Better use of random number generator bits?

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2006-03/msg01128.html>

---

- *From:* David Bernier <david250@xxxxxxxxxxxx>
  - *Date:* Mon, 06 Mar 2006 21:57:05 -0500
- 

DAC wrote:

This is important for us because for example to shuffle a deck of cards we are taking 51 random numbers (between 0 and 51 down to between 0 and 1). This on average is a total of about 2880 bits to shuffle a deck (of course there is no maximum on this number and I have encountered one situation that took over 13,000 bits), where you should really only need  $\log_2(52!) = 226$  bits to specify a specific shuffle. So at the moment we are using on average 10 times as many bits as we really need – and quantum generator modules are expensive and soon we will need the numbers at a rate faster than we can produce them.

Suppose 10 cards have been chosen. Then, 42 remain so one wants to choose a random number between 0 and 41 inclusive.

4,294,967,292 is a multiple of 42; also, it is less than  $2^{32}$  so it can be represented as a 32-bit unsigned integer.

Let's choose a random 32-bit unsigned integer and check if it is less than 4,294,967,292, and then

- accept it if it is less than 4,294,967,292 and
- reject it otherwise.

We repeat until such a number is found, and then take the remainder upon dividing by 42. The result is a random number chosen uniformly between 0 and 41.

These large multiples of 52, 51, ... 2 can be computed and stored in an array.

In the above, case (b) will occur very rarely...

I wrote a program in C to illustrate the idea and hopefully it is not too badly written.

Of course, the basic idea could be used with 16-bit integers or 8-bit integers. If the objective is to waste the least number of

## Re: Better use of random number genator bits?

random bits on average, I don't know which range of integers is best, 8-bit, 9-bit, ...?

David Bernier

/\*\*

This program is based on RNG's developed by G. Marsaglia.  
see:

<http://groups.google.com/group/sci.stat.math/msg/b6e5b69842772ad4/>

[sci.math and sci.stat.math, Jan 20 1999]

NOTE: #define SHR3... has to be on 1 line

\*\*\*/  
  
#include <stdio.h>  
#define NDECKS 10  
  
/\* Marsaglia rng \*/  
  
#define znew (zrand=36969\*(zrand&65535)+(zrand>>16))  
#define wnew (wrand=18000\*(wrand&65535)+(wrand>>16))  
#define MWC ((znew<<16)+wnew )  
#define SHR3 (jsr\_rand^=(jsr\_rand<<17),  
jsr\_rand^=(jsr\_rand>>13),jsr\_rand^=(jsr\_rand<<5))  
#define CONG (jcongrand=69069\*jcongrand+1234567)  
#define KISS ((MWC^CONG)+SHR3)  
  
/\*\* NOTE: This program is not reseeded at startup! \*\*\*/  
  
static unsigned long zrand=362436069, wrand=521288629;  
static unsigned long jsr\_rand=123456789, jcongrand=380116160;  
int main()  
{  
long deck\_number;  
unsigned long max\_array[51];  
int i, j, n;  
unsigned long n\_cast\_as\_UL;  
int a\_deck[52];  
int step;  
int card\_selected;  
unsigned long randnum\_UL;  
  
/\*\* Begin: prepare array of 51 maximum

Re: Better use of random number genator bits?

## Re: Better use of random number genator bits?

```
unsigned long int values ***/

for(i=0;i<51;i++)
{
n = 52 - i;
n_cast_as_UL = (unsigned long) n;
max_array[i] = (4294967295UL/n_cast_as_UL)*n_cast_as_UL;
/**/ roughly, max_array[i] is
some large integer multiple of (52-i) ***/
}

/**/ End: prepare array of 51 maximum
unsigned long int values ***/

for(i=0;i<51;i++)
{
printf("%u ", max_array[i]);
if( 5 == (i%6))
{
printf("\n");
}
}
printf("\n\n");

for(deck_number=0;deck_number<NDECKS;deck_number++)
{
printf("shuffling deck number %ld...\n", deck_number);

/**/ shuffle a new deck ***/
for(step=0;step<51;step++)
{
card_selected=0;
while(card_selected == 0)
{
randnum_UL = KISS;
if(randnum_UL < max_array[step])
{
a_deck[step] = (int)(randnum_UL%((unsigned long) 52-step));

/**/ we are done with selection number 'step' ***/
card_selected=1;
}

/**/ if we end up here, we are not done
with selection number 'step' ***/

} /**/ brace closes last while... ***/

/**/ Done: selection number 'step' ***/
} /**/ closes last for ... ***/
```

## Re: Better use of random number genator bits?

```
a_deck[51] = 0;

/** Deck is now shuffled */

for(j=0;j<52;j++)
{
printf("%2d ", a_deck[j]);
if( 12 == (j%13))
{
printf("\n");
}
}

} /** brace closes 'for(deck_number=0[...] ' */

return 0;
}
.
```