Subject: Re: Randomize array order
Posted by Paolo Grigis on Fri, 27 Jul 2007 13:59:08 GMT
View Forum Message <> Reply to Message

Allan Whiteford wrote:
> Conor wrote:
>> Hi everyone!
>>
>>   Anyone know an efficient way to randomize an array (I have a
>> sorted array that I want unsorted). Initially, I tried something like
>> this:
>>
>> array = findgen(1000000)
>> unsort = array[sort(randomu(seed,1000000))]
>>
>> It works, but sorting on a million elements is rather slow. Anyone
>> know a faster way?
>>
>
> Slightly different point and probably a bit academic:
>
> If you have a million elements then you have 1000000! (i.e. one million
> factorial) different ways to re-order the data. However, your seed is a
> 4 byte integer which can only take 2^32 different values.
>
> Some messing about suggests that:
>
> 1000000! =~ 10^5568636
>
> which means there are ~ 10^5568636 different ways to re-arrange your
> elements as opposed to the 4 x 10^9 values your seed can take.
>
> Thus, using any of the algorithms suggested you're only going to sample
>
>   10^-5568625 %
>
> of the possible values. This is a really small number. It means that no
> matter how hard you try and how many times you do things you'll never be
> able to access anything but a tiny number of the possibilities without
> doing multiple shufflings - I think it's something like 618737
> sub-shufflings (i.e. 5568636 / 9) but that could be wrong. However, that
> requires producing 618737 seeds per major-shuffle (and you can't use a
> generator based on a 4 byte seed to produce these seeds).
>
> But, since you're only going to be running the code 1000-10,000 times
> (which is much smaller than 4e9)

I am by no means an expert, but I think that in general, common sense

suggests not to use more random numbers for one project than the cycle
length of the generator (that is, the length at which the numbers start
to repeat themselves). From the docs is hard to guess how long the cycle
is, but it can be at most 2^32 for a generator using long ints. So I
wouldn't suggest doing more than 2^32/1d6 runs of the code, if one is
using 1d6 numbers per run.

If more random deviates are needed, it would be a good idea to use a
random generator with a longer cycle.

Ciao,
Paolo

I guess everything will be ok. I don't
> know if anyone has studied possible correlations of results as a
> function of the very small number of seeds (compared to the data),
> whatever random number generator is used and the shuffling method.
> Presumably they have and presumably everything is ok. Does anyone know?

>
> Thanks,
>
> Allan