Subject: Re: Randomize array order
Posted by Vince Hradil on Thu, 26 Jul 2007 14:58:19 GMT
View Forum Message <> Reply to Message

On Jul 26, 8:40 am, Conor <cmanc...@gmail.com> wrote:
> On Jul 26, 9:30 am, Allan Whiteford
>
>
>
> <allan.rem...@phys.remove.strath.ac.remove.uk> 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?
>
>> Conor,
>
>> Is it a million elements you want to do?
>
>> The following scales better:
>
>> pro shuffle,in
>> b=long(n_elements(in)*randomu(seed,n_elements(in)))
>> for i=0l,n_elements(in)-1 do begin
>> tmp=in[i]
>> in[i]=in[b[i]]
>> in[b[i]]=tmp
>> end
>> end
>
>> but on my machine, a million elements is around about where it starts to
>> become as efficient as yours. For 10 million elements the above is a bit
>> (17.05 seconds vs 12.92 seconds) but for 1 million elements they both
>> come in at around 1.2 seconds (1.15 seconds vs 1.26 seconds). The above
>> will scale as pretty much O(n) since it doesn't do any sorting but it
>> takes a hit in the practical implementation because of the loop in
>> IDL-space. Your suggestion will scale worse than O(n) but it seems the
>> overlap in the two methods is exactly where you want to work.
>
>> Maybe my loop can be made more efficient in practical terms but I don't

>> think this is any better algorithm in terms of scaling (hard to imagine
>> anything that could go faster than O(n) to randomise n things).
>
>> Probably not helpful but I thought it was interesting that the
>> cross-over is exactly where you want to work. But, maybe I should get
>> out more if I think that's especially interesting.
>
>> Thanks,
>
>> Allan
>
> Thanks for the suggestions guys! I'll have to play around and see
> what works best.

Here's a table of results from my machine. All times are in seconds.
PC single processor, WinXP, IDL6.4

i	Niter	Rand-meth	Loop-meth
0	100000	0.0929999	0.110000
1	166810	0.0779998	0.0940001
2	278256	0.140000	0.157000
3	464158	0.297000	0.297000
4	774263	0.578000	0.562000
5	1291549	1.09400	0.890000
6	2154435	2.06300	1.48400
7	3593812	3.84400	2.56300
8	5994841	7.09400	4.31300
9	10000000	13.0470	7.29800

Here's my code:

```
function runsort, array
  na = n_elements(array)
  return, array[sort(randomu(seed,na))]
end

function lunsort, array
  na = n_elements(array)
  rarray = array

  b = long(na*randomu(seed,na))
  for i=0l, na-1 do begin
    tmp = rarray[i]
    rarray[i] = rarray[b[i]]
    rarray[b[i]] = tmp
  endfor

  return, rarray
```

```
end

pro test_unsort, randi=randi, loopi=loopi, nel=nel

  n = 10l
  nlo = 5l
  nhi = 7l
  fndx = findgen(n)/float(n-1)
  nel = long(10^( (nhi-nlo)*fndx + nlo ) )

  randi = fltarr(n)
  loopi = fltarr(n)
  for i=0l, n-1 do begin
    array = findgen(nel[i])
    t = systime(1)
    unsort = runsort(array)
    randi[i] = systime(1)-t

    t = systime(1)
    unsort = lunsort(array)
    loopi[i] = systime(1)-t

    print, i, nel[i], randi[i], loopi[i]
  endfor

  return
end
```