Subject: Re: Randomize array order
Posted by Allan Whiteford on Thu, 26 Jul 2007 13:30:00 GMT
View Forum Message <> Reply to Message

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