Subject: Re: how to sort data based on other sorted data
Posted by Tom McGlynn on Mon, 14 Jan 2008 05:10:24 GMT
View Forum Message <> Reply to Message

On Jan 12, 2:35 pm, Craig Markwardt
<craigm...@REMOVEcow.physics.wisc.edu> wrote:
> Tom McGlynn <t...@milkyway.gsfc.nasa.gov> writes:
>> On Jan 10, 3:51 pm, placebo <willie.mad...@gmail.com> wrote:
>>>> try Craig Markwardt's multisort
>
>>>> http://astrog.physics.wisc.edu/~craigm/idl/arrays.html#MULTI SORT
>
>>> Brian,
>
>>> The multisort method works quite well.
>
>> ...
>
>> Multisort works fine and knowing Craig will work very robustly
>> but I don't think you need to have any limit on
>> the number of columns. Below is a routine that should
>> be able to handle an arbitrary number of columns and rows...
>
> Hi Tom, thanks for the interesting contribution. I think the essence
> of BIGSORT and MULTISORT are similar, namely to build up a surrogate
> sort key. MULTISORT does it with strings and BIGSORT does it with
> integers. Clearly sorting by integers will be faster.
>
> Incidentally, the MULTISORT limitation of ten sort keys was purely
> arbitrary, and it would be trivial to extend. But honestly, who would
> need so many sort keys? It's like the people who shop for cameras
> these days based on megapixels...
>
> Craig

Hi Craig,

When I was writing the program originally I think I would have agreed
that both programs were building proxy keys but on reflection I think
that what Bigsort is doing is better described as iterating over
each column and refining a partial ordering until it either gets a
total
ordering or runs out of columns. It not obvious which should be
faster
Multisort has to build a string for each row, but does only a
single sort. Bigsort keeps just a single integer for each row, but it
may sort
as many as 2*nCol times (once over the values of each column and once

per column
over the current sort index) If all the columns were strings, I
wouldn't
be surprised if Multisort was faster and it's close enough that it may
be implementation dependent in any case.

On the question of whether the column limits matter:
When we limit sorting to cases where we were preparing output or other
cases
where we were concerned with the actual order, then I agree that we
rarely
need to worry about more than a few columns. However I can imagine
using
a canonical ordering to answer questions like: Are there any duplicate
rows? What are they? How many distinct rows are there? While there
may be
other ways to answer these, being able to simply sort rows without
worrying
about the details of the columns makes answering this kind of question
trivial. For this kind of use having something that doesn't have a
specific
limit on columns seems desirable.

Tom
