Subject: Re: Sort without loops
Posted by JD Smith on Wed, 17 Aug 2005 19:02:57 GMT
View Forum Message <> Reply to Message

On Wed, 17 Aug 2005 10:08:07 +0100, Ian Dean wrote:

> Hi All,
> I have a large string array (~100000 elements) that need sorting on two
> fields within each string.
>
> e.g. array=['F;100', 'ABC;6', 'DE;2', 'DE;10', 'DE;1']
>
> Order required is a) sort items to left of ';' followed by b) sort items
> numerically to right of ';'
> This would produce:
> ABC;6 DE;1 DE;2 DE;10 F;100
>
> A simple sort (sort(array)) procudes:
> ABC;6 DE;1 DE;10 DE;2 F;100
>
> The only way I've found is to conmvert the RH part to I4.4 format within a
> loop and search on the new values:
> ......
> for j=0, n_elements(array)-1 do begin
> parts=strsplit(array[j], ';', /extract)
> RH=string(fix(parts[1]), format='(I4.4)')
> new[j]=parts[0]+RH
> endfor
> order=sort(new)
> ...
> i.e
> new array is F;0100 ABC;0006 DE;0002 DE;0010 DE;0001
> which is then sorted correctly.
>
> Is there a clever way of sorting on two fields like this without using a
> loop. The above algorithm is faaaar slower than just using sort.

Similar to your approach, and Peter's, but without the loop:

```
IDL> pos=transpose(strpos(array,';'))
IDL>  s=sort(strmid(array,0,pos)+string(FORMAT='(I5.5)',strmid(arr ay,pos+1)))
```

IDL has no good way to alter the sorting semantics, to simultaneously
sort on multiple fields. Most languages offer the ability to specify
a sorting function, which compares two elements for GT, LT, or EQ,
using any logic you like. Since IDL doesn't allow this, you're forced
to re-cast your entire set as strings or integers.

JD