Subject: Re: Sort without loops
Posted by peter.albert@gmx.de on Wed, 17 Aug 2005 10:36:30 GMT
View Forum Message <> Reply to Message

Hi Ian,

> 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.

I don't know how to sort over two fields at the same time, but I could propose a much faster way ofcreating you new string array, avoiding the for loop:

```
spos = strpos(array, ";")

left = strmid($
   array, $
   transpose(replicate(0, n_elements(array))), $
   transpose(spos)$
   )

right = string($
   fix($
      strmid($
         array, $
         transpose(spos+1)),$
      ), $
      format = '(i4.4)'$
   )
```

The basic idea is to use an array for the variables "first_character" and "length" in the call to strmid. It has to be the transposed ones, as the first dimension of those arrays determines how many substrings are extracted from each element of the string array - one in your case.

On my PC this method is approx. 10 times faster as the for-loop. If it is still too slow, one could think of providing more sophisticated arrays for first_character and length with a 2 as first dimension, i.e. extracting both substrings in one go. But I doubt that this would really help a lot.

Best regards,

Peter

Subject: Re: Sort without loops
Posted by R.Bauer on Wed, 17 Aug 2005 14:24:28 GMT
View Forum Message <> Reply to Message

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.
>
> I hope I have made this as clear as mud.#
>
> In expectation,
> Ian
>
>

probably n_sort is what you want.

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/n_sort_dbase.pro.html

cheers
Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======

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
