Subject: Re: sorting column arrays
Posted by greg michael on Fri, 30 Jun 2006 12:30:29 GMT
View Forum Message <> Reply to Message

How about this (no loop):

```
v=lonarr(4,n_elements(array))
reads,strmid(array,4),v
print,v[*,sort(v[1,*]+v[2,*]/10.+v[0,*]/100.)]
```

The last line makes the sort in one go, since SORT doesn't guarantee anything about the order of identical elements. You might have to adjust the coefficients if your data has a wider range than the sample you show.

regards,
Greg

Subject: Re: sorting column arrays
Posted by cgguido on Fri, 30 Jun 2006 16:15:19 GMT
View Forum Message <> Reply to Message

> The last line makes the sort in one go, since SORT doesn't guarantee
> anything about the order of identical elements. You might have to
> adjust the coefficients if your data has a wider range than the sample
> you show.

sort does indeed tend to jumble things up a little, but bsort does the job right:
http://www.dfanning.com/tips/sort.html.

also, something like this should work....

```
IDL> d = $
IDL> [[4,  7, 0, 1], $
IDL>  [3,  6,  2,  9],$
IDL>  [2, 2, 0,  2],$
IDL>  [42,  4,  2,  7],$
IDL>  [256,  3, 0,  2],$
IDL>  [34,  2,  1,  5]]
IDL>
IDL> s1 = bsort(d[1, *])
IDL> s2 = bsort(d[3, s1])
IDL> s3 = bsort(d[2, s1[s2]])
IDL> sd = d[*, s1[s2[s3]]]
IDL>
IDL> print,  d,  'sorted by columns 2>3>1', sd
```

```
     4       7       0       1
     3       6       2       9
     2       2       0       2
    42       4       2       7
   256       3       0       2
    34       2       1       5
sorted by columns 2>3>1
     4       7       0       1
     2       2       0       2
   256       3       0       2
    34       2       1       5
    42       4       2       7
     3       6       2       9
```

Gianguido
