Subject: Multi-column sort
Posted by Percy Pugwash on Thu, 01 Mar 2012 15:02:06 GMT
View Forum Message <> Reply to Message

I would like to sort a 2D string by column, specifying primary,
secondary and further sort criteria (i.e. rows which are equal based
on criterion 1 are sorted by criterion 2).

Is there any neat way to do this in IDL?

I'd thought of using the following:

```
tosort = [3,1,5] ; Sort by column 3, then 1, then 5
maxlen = max(strlen(sortarray[tosort,*])) ; Length of longest string
paddedarray =
 string(sortarray[tosort,*],format='(a'+string(maxlen,format= '(i0)') +
')' ) ; Pad all strings to match longest length
concatarray = paddedarray[0,*]
for i = 0, n_elements(tosort)-1 do concatarray += paddedarray[i,*] ;
Concatenate strings across columns
indices = sort(concatarray) ; Sort the concatenated strings
```

However, this method does not allow me to specify which direction the
sort should go for each of the sort columns. Can anyone think of a way
to extend the method to allow this (or a completely different method
which achieves the same effect!)?

Thanks,

P

Subject: Re: Multi-column sort
Posted by wlandsman on Thu, 01 Mar 2012 22:02:08 GMT
View Forum Message <> Reply to Message

You might look at Craig Markwardt's multisort

http://cow.physics.wisc.edu/~craigm/idl/down/multisort.pro

On Thursday, March 1, 2012 10:02:06 AM UTC-5, Percy Pugwash wrote:
> I would like to sort a 2D string by column, specifying primary,
> secondary and further sort criteria (i.e. rows which are equal based
> on criterion 1 are sorted by criterion 2).
>
> Is there any neat way to do this in IDL?
>
> I'd thought of using the following:

>
> tosort = [3,1,5] ; Sort by column 3, then 1, then 5
> maxlen = max(strlen(sortarray[tosort,*])) ; Length of longest string
> paddedarray =
> string(sortarray[tosort,*],format='(a'+string(maxlen,format= '(i0)') +
> ')') ; Pad all strings to match longest length
> concatarray = paddedarray[0,*]
> for i = 0, n_elements(tosort)-1 do concatarray += paddedarray[i,*] ;
> Concatenate strings across columns
> indices = sort(concatarray) ; Sort the concatenated strings
>
> However, this method does not allow me to specify which direction the
> sort should go for each of the sort columns. Can anyone think of a way
> to extend the method to allow this (or a completely different method
> which achieves the same effect!)?
>
> Thanks,
>
> P

Subject: Multi-column sort
Posted by cgguido on Fri, 02 Mar 2012 12:26:00 GMT
View Forum Message <> Reply to Message

Firstly, sort() does not maintain the order of identical elements so I'd use bsort() which you can find online somewhere, can't remember where... I believe it has a /reverse or /invert option, not at my computer right now.

Secondly, you should bsort columns in increasing order of importance, with the most important sort last.

I have a dumb for-loop procedure that does that. This requires multiple searches through the array, which might not be optimal, but once you write it, you'll never use sort instead of bsort again :-)

G

Subject: Re: Multi-column sort
Posted by Percy Pugwash on Mon, 05 Mar 2012 10:53:17 GMT
View Forum Message <> Reply to Message

Thanks very much. I had looked at Craig Markwardt's multisort, but it didn't quite do what I wanted (the number of columns to sort by was limited and could not easily be changed at run-time).

I've not looked at bsort yet, but will have a look, thanks. My only concern is that I'm aware that

bubble sort is usually much slower than quicksort, which I believe sort() uses... Might still be worth it though.

The solution I came up with is below, in case anyone is interested. Please bare in mind that it has not been properly tested, but seems to be working.

P

```
function sort_strcolumns, strtable, indices
  maxlen = max(strlen(strtable[abs(indices),*]))
  ncols = n_elements(indices)
  nrows = (size(strtable,/dim))[1]
  sortlist =  reform(string(strtable[abs(indices),*],f='(a-'+string(maxlen
,f='(i0)')+')'),ncols,nrows,/overwrite)
  sortlist = reform(byte(sortlist),maxlen*ncols,nrows,/overwrite)
  for i=0,n_elements(indices)*maxlen-1 do sortlist[i,*] *= (-1)^(indices[i/maxlen] lt 0)
  return, sort(string(sortlist))
end
```

On Friday, 2 March 2012 12:26:00 UTC, Gianguido Cianci wrote:
> Firstly, sort() does not maintain the order of identical elements so I'd use bsort() which you can find online somewhere, can't remember where... I believe it has a /reverse or /invert option, not at my computer right now.
>
> Secondly, you should bsort columns in increasing order of importance, with the most important sort last.
>
> I have a dumb for-loop procedure that does that. This requires multiple searches through the array, which might not be optimal, but once you write it, you'll never use sort instead of bsort again :-)
>
> G

Subject: Re: Multi-column sort
Posted by Percy Pugwash on Mon, 05 Mar 2012 10:59:21 GMT
View Forum Message <> Reply to Message

**bear in mind.

Oh dear...

P

On Monday, 5 March 2012 10:53:17 UTC, Percy Pugwash wrote:
> Thanks very much. I had looked at Craig Markwardt's multisort, but it didn't quite do what I wanted (the number of columns to sort by was limited and could not easily be changed at run-time).

>
> I've not looked at bsort yet, but will have a look, thanks. My only concern is that I'm aware that bubble sort is usually much slower than quicksort, which I believe sort() uses... Might still be worth it though.
>
> The solution I came up with is below, in case anyone is interested. Please bare in mind that it has not been properly tested, but seems to be working.
>
> P
>
> function sort_strcolumns, strtable, indices
> maxlen = max(strlen(strtable[abs(indices),*]))
> ncols = n_elements(indices)
> nrows = (size(strtable,/dim))[1]
> sortlist = reform(string(strtable[abs(indices),*],f='(a-'+string(maxlen
,f='(i0)')+')'),ncols,nrows,/overwrite)
> sortlist = reform(byte(sortlist),maxlen*ncols,nrows,/overwrite)
> for i=0,n_elements(indices)*maxlen-1 do sortlist[i,*] *= (-1)^(indices[i/maxlen] lt 0)
> return, sort(string(sortlist))
> end
>
> On Friday, 2 March 2012 12:26:00 UTC, Gianguido Cianci wrote:
>> Firstly, sort() does not maintain the order of identical elements so I'd use bsort() which you can find online somewhere, can't remember where... I believe it has a /reverse or /invert option, not at my computer right now.
>>
>> Secondly, you should bsort columns in increasing order of importance, with the most important sort last.
>>
>> I have a dumb for-loop procedure that does that. This requires multiple searches through the array, which might not be optimal, but once you write it, you'll never use sort instead of bsort again :-)
>>
>> G

Subject: Re: Multi-column sort
Posted by Brian Wolven on Mon, 05 Mar 2012 17:54:21 GMT
View Forum Message <> Reply to Message

On Monday, March 5, 2012 5:59:21 AM UTC-5, Percy Pugwash wrote:
> **bear in mind.
>
> Oh dear...

Yeah, we're not allowed to do the other at work.
