
Subject: sorting string arrays - non alphabetic and user defined order

Posted by [rkombiyil](#) on Mon, 16 Oct 2006 10:34:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Fellow Uber-idlers,

I am terrified/mortified/petrified and stupified by this problem.. I am too lazy and try to do as much text processing and unixing from within IDL .. with as little user intervention as possible :-)

Maybe this is too obvious for you ubergeeks, I guess my greycells don't have many pathways...

--

Say, I have a list of names stored in this way:

```
namelist=['Daddy','Groggy','Ally','Curry','Emmy','Bully','Jockey','Hippy','Itchy','Fluffy']
```

I have another list of names, which

1. may/may not contain all the names in the above list.
2. and this list is ouput (and alphabetically sorted by default) from IDL. But I want to sort this array the way I want, for example, comparing to 'namelist'.

For example: IDL sorted list is:

```
mylist=['Emmy','Fluffy','Itchy','Jockey']
```

I want to reorder 'mylist' comparing to 'namelist' to ['Emmy','Jockey','Itchy','Fluffy'] and also reset the indices to [0,1,2,3] instead of [4,9,8,6]

I hope I made sense.. That is

```
* take 'mylist'  
** compare with 'namelist'  
*** order names in 'mylist' in the same way as in 'namelist' accounting  
for missing names  
**** reset indices starting from 0 in the reordered list
```

--

I tried couple of different permutations and combinations using stregex and such, but I screw up when it comes to resetting indices in the reordered list :(I think pointers is the way to go, but I haven't learned pointers in IDL.

Any suggestions or advice appreciated!

Thanks in advance,

~rk

Subject: Re: sorting string arrays - non alphabetic and user defined order

Posted by [greg michael](#) on Tue, 17 Oct 2006 07:14:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

It seemed much more fun when I thought we were talking about dwarves...

I'm not sure I understnad what you're asking in the new post. It seems to me that in my code 'namelist' is your NamelistB, and mylist is NamelistC. NamelistA is superfluous to the problem, isn't it?

Here's a modified code to show what I mean - NamelistA has a couple of extra entries, but I never use it. Even if they are present in mylist (NamelistC), they will be ignored when filtered with mylist (NamelistB). Even a double entry will turn up only once.

```
namelistA=['Sneezy','Daddy','Groggy','Dopey','Ally','Curry',  
'Emmy','Bully','Jockey','Hippy','Itchy','Fluffy']
```

```
namelist=['Daddy','Groggy','Ally','Curry','Emmy','Bully','Jo ckey','Hippy','Itchy','Fluffy']  
mylist=['Emmy','Fluffy','Itchy','Jockey','Groggy','Groggy',' Sneezy']  
n=n_elements(namelist)  
m=n_elements(mylist)  
a1=rebin(transpose(indgen(m)),n,m)  
a2=rebin(indgen(n),n,m)  
s=mylist[a1] eq namelist[a2]  
result=namelist[where(total(s,2) gt 0)]
```

```
IDL> print,result
```

```
Groggy Emmy Jockey Itchy Fluffy
```

'result' will automatically have the right size and be indexed from zero upwards - you don't need to worry about declaring it beforehand. Or have I missed what you're asking?

regards,
Greg

metachronist wrote:

> Hi Greg,

>

> Thank you for your quick response, it is much appreciated. Your method

> works fine. I am afraid I left out of couple of questions related to

> the problem.
>
> Is there a way to dynamically allocate arrays (varying size) ?
> Specifically, this is what I am trying to do.
>
> #1
>
> I have a station database. This "string" array contains details
> pertaining to all the stations (names,locations,code etc.). Let's call
> this NAMELIST 'A'
>
> #2
>
> Now, I made a list of stations I want to look at from the above
> database. Let's call this NAMELIST 'B'
>
> #3
>
> The list of stations provided to me (NAMELIST 'C') may or may not
> contain all the stations in namelist 'B'
>
> #4
>
> I make a 1-1 string match between namelists 'B' and 'C' and extract
> only those stations that are present in B and C, and order C similar to
> B (user defined non-alphabetic)
>
> #5
>
> Problem is: I know the dimensions of the string array 'B' because I get
> to choose the stations I want from the original big database. Now, I
> don't know the dimensions of 'C' because it is variable, it may have
> same dimensions as B or less than B or none at all and the order might
> vary. Since, it can "ALSO" have same dimensions of B (max possible
> dimension), I define
>
> C=strarr(n_elements(B))
>
> But when the dimensions are less than that in B, there are elements
> with 'strlen' equal to zero. For example, C[3] maybe empty and may be
> C[7] and all other elements of the array may be filled.. I want to get
> rid of these trailing/beginning/in between empty (zero length) elements
> (I know this is because of the above declaration)
> and make this array to have dimensions = # of non-zero elements
>
> Is there a way to tackle such situations? I tried to index and
> increment the # of non-zero elements and redeclare the dimensions of C
> to be # of non-zero elements.. But it didn't work.. Meanwhile, I will

> try to modify your code to see if it works for my need.
>
> I appreciate your time and help!
> Thanks much,
> ~rk

Subject: Re: sorting string arrays - non alphabetic and user defined order
Posted by [rkombiyil](#) on Tue, 17 Oct 2006 07:56:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, I agree dwarves are more fun :-)

NamelistA is definitely "superfluous" to the problem. That was just for clarification's sake.. but I think I didn't do a good job.. Let me try one last time.

--

I don't know 'mylist' beforehand.

To start with, I have a bunch of random names and I read them into an array(say, 'givenlist').

Then I compare 'givenlist' with 'namelist' using "strmatch" and obtain another 'array' (say, 'mylist') ordered in the same way as given in 'namelist'. Since I don't know the dimensions of 'mylist', (it varies based on number of elements that are common but it can contain all the names in 'namelist') I declared the dimension of 'mylist' to be that of 'namelist'.

Tho I can get correct match sorted in the way I want, if some names are missing from 'namelist',

say,

namelist=

```
['Daddy','Groggy','Ally','Curry','Emmy','Bully','Jockey','Hi ppy','Itchy','Fluffy']
```

```
mylist= ['Groggy', 'Emmy', 'Jockey', 'Itchy', 'Fluffy']
```

then the dimensions of 'mylist' are still n_elements(namelist) and
strlen(mylist[0]) = 0 ,strlen(mylist[3])=0 etc.

Hence, I was thinking the best way to tackle such a situation would be to dynamically allocate the size of 'mylist' depending on how many matches are found between 'namelist' and 'givenlist' .. I am sorry if I am still not clear and messed up in explaining what I was trying to do. Maybe it needn't be complicated like this. But I guess I would have to learn from experiance.

Thanks for taking time off to reply,

~rk

Subject: Re: sorting string arrays - non alphabetic and user defined order

Posted by [greg michael](#) on Tue, 17 Oct 2006 11:43:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

hmm... it seems to me that 'result' is the array you are looking for: it has the elements in the order you want, and they have the indices numbered from zero (i.e. there are no empty elements). I don't see why you would want the dimensions of mylist or result to be the same as namelist. mylist is given to you, and has its own size - the code will work whatever its size; 'result' is calculated and has its own 'dynamically allocated' size.

If you're sure you want an output with blanks - although I can't see how you would use it - try tacking this on the end:

```
result2=namelist  
result2[where(total(s,2) eq 0)]=''
```

```
IDL> print,'#'+result2+'#'  
## #Groggy# ## ## #Emmy# ## #Jockey# ## #ltchy# #Fluffy#
```

So you have namelist with the unneeded elements set to empty.

If I still haven't got it, perhaps you could write what you would expect the output to look like for a couple of cases.

regards,
Greg

Subject: Re: sorting string arrays - non alphabetic and user defined order

Posted by [David Fanning](#) on Tue, 17 Oct 2006 12:00:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

Greg Michael writes:

```
> hmm... it seems to me that 'result' is the array you are looking for:  
> it has the elements in the order you want, and they have the indices  
> numbered from zero (i.e. there are no empty elements).
```

It seems that way to me, too. But I had to work with the solution for a time before I understood it. It would have taken me a LONG time to think of using indices rather than strings to come up with a solution. Perhaps the solution was TOO concise. I've expanded on it just a little bit in this article:

http://www.dfanning.com/idl_way/strsort.html

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: sorting string arrays - non alphabetic and user defined order
Posted by [rkombiyil](#) on Mon, 23 Oct 2006 00:53:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Greg,

Just wanted to let you know that I modified your code to suit my needs.

It is an elegant method and I haven't thought of it. Cool. Dimensional

jugglery, as Dr.Fanning puts it, is something I need to learn well

besides learning how to minimize *for* loops in my codes :) I guess

those come with experience as I learn IDL better.

Thanks for helping,

/metachronist

Subject: Re: sorting string arrays - non alphabetic and user defined order
Posted by [JD Smith](#) on Mon, 23 Oct 2006 21:03:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Tue, 17 Oct 2006 06:00:03 -0600, David Fanning wrote:

> Greg Michael writes:

>

>> hmm... it seems to me that 'result' is the array you are looking for:

>> it has the elements in the order you want, and they have the indices

>> numbered from zero (i.e. there are no empty elements).

>

> It seems that way to me, too. But I had to work with

> the solution for a time before I understood it. It

> would have taken me a LONG time to think of using

> indices rather than strings to come up with a solution.

> Perhaps the solution was TOO concise. I've expanded on

> it just a little bit in this article:

>

> http://www.dfanning.com/idl_way/strsort.html

This is similar to the standard inflate and compare WHERE_ARRAY method. So it would be somewhat simpler (if a bit slower) just to say:

```
mylist=mylist[sort(where_array(mylist,namelist,/PRESERVE_ORDER))]
```

see turtle.as.arizona.edu/idl/where_array.pro for the modified version which includes PRESERVE_ORDER.

That said, this exhibits the classic defect of "scale-em-up-and-compare" methods which the REBIN/REFORM stuff enables: it starts to get ugly when your comparison vectors get long, scaling as the product of their lengths, and gobbling up enormous amounts of memory in the process. We discuss in detail the pros and cons of the various methods here:

http://www.dfanning.com/tips/set_operations.html

For long vectors, you'll be better off with a sort-based algorithm (e.g. ind_int_SORT, see above):

```
mylist=mylist[sort(ind_int_sort(namelist,mylist))]
```

JD

Subject: Re: sorting string arrays - non alphabetic and user defined order

Posted by [greg michael](#) on Tue, 24 Oct 2006 07:39:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Agreed - you wouldn't want to apply the method I proposed to very long lists - sorting is the real solution...

Greg
