Subject: Re: error with sort
Posted by davidf on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

R.Bauer (R.Bauer@fz-juelich.de) writes:

> Dear David,
>
> that's all right, but why get I different results on different Operation
> Systems?

I'm reliably informed that you get different results because
IDL uses the system supplied qsort() routine on each platform,
and as they are different implementations, they are free to
return different sub-orders for the identical elements.

But, as I say, I can't see how it makes much difference,
although I did appreciate Wayne Landsman's perspective.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: error with sort
Posted by Liam E. Gumley on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

"R.Bauer" wrote:
> SORT() is not platform independent!!!!!

So if all elements of the array are identical, you wish to maintain the
index order. How about this:

```
if (n_elements(uniq(arr)) eq 1) then $
  idx = lindgen(n_elements(arr)) else $
  idx = sort(arr)
```

Cheers,
Liam.

--
Liam E. Gumley
Space Science and Engineering Center, UW-Madison
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: error with sort
Posted by landsman on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

If you are worried about what SORT does to equal values, you might
instead want to use the program bsort.pro, available at

ftp://idlastro.gsfc.nasa.gov/pub/pro/misc/bsort.pro

This procedure ensures that equal values are always maintained in the
initial order, i.e.

```
IDL> print,bsort([1,1,1,1,1])
       0       1       2       3       4
```

One place where this procedure is useful is when you are sorting on more
than one parameter. For example, suppose you have a set of
temperature and pressure measurements (T, P), and want the primary sort
to be by temperature -- but whenever temperatures are equal, you want
the values sorted by pressure. One can do this as follows:

```
i1 = sort(P)            ;Secondary sort on pressure
i2 = bsort(T[i1])       ;primary sort on temperature
```

and i2 will give the desired indexing.
Of course, BSORT will be slower than the intrinsic SORT function.

Wayne Landsman landsman@mpb.gsfc.nasa.gov

Subject: Re: error with sort
Posted by davidf on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

R.Bauer (R.Bauer@fz-juelich.de) writes:

> Dear David,

>
> that's all right, but why get I different results on different Operation
> Systems?

I don't know. But I am scratching my head trying to
think why it would make one bit of difference. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: error with sort
Posted by R.Bauer on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> R.Bauer (R.Bauer@fz-juelich.de) writes:
>
>> print,sort([1,1,1,1,1])
>
> I'm pretty sure that no matter how you would
> order the sort of these numbers, Reimer, you
> would get the same answer. :-)
>
> And, really, that's all these different numbers
> are saying. It doesn't matter where you *start*
> the sort, when values are the same, the order is
> irrelevant. The SORT function is obviously selecting
> a random number as the starting point of the sort.
> (In other words, to sort something you pick a
> number and then the next number is greater than
> this number, or less than this number, etc. If the
> number happens to be *equal* to the pick. Just
> put it here, it's order doesn't matter. That's
> what you are seeing in the pattern (or non-pattern)
> you are detecting. SORTs are more efficient, in
> general I think, when they use random numbers
> to pick starting locations.
>

> Cheers,
>
> David
>

Dear David,

that's all right, but why get I different results on different Operation Systems?

Reimar

File Attachments

1) R.Bauer.vcf, downloaded 98 times

Subject: Re: error with sort
Posted by davidf on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

R.Bauer (R.Bauer@fz-juelich.de) writes:

> print,sort([1,1,1,1,1])

I'm pretty sure that no matter how you would order the sort of these numbers, Reimer, you would get the same answer. :-)

And, really, that's all these different numbers are saying. It doesn't matter where you *start* the sort, when values are the same, the order is irrelevant. The SORT function is obviously selecting a random number as the starting point of the sort. (In other words, to sort something you pick a number and then the next number is greater than this number, or less than this number, etc. If the number happens to be *equal* to the pick. Just put it here, it's order doesn't matter. That's what you are seeing in the pattern (or non-pattern) you are detecting. SORTs are more efficient, in general I think, when they use random numbers to pick starting locations.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: error with sort
Posted by R.Bauer on Mon, 10 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

```
print,sort([1,1,1,1,1])
       0       1       2       3       4
help,!version,/struc
** Structure !VERSION, 5 tags, length=40:
   ARCH            STRING    'x86'
   OS              STRING    'linux'
   OS_FAMILY       STRING    'unix'
   RELEASE         STRING    '5.2.1'
   BUILD_DATE      STRING    'Jun  4 1999'
```

File Attachments
1) R.Bauer.vcf, downloaded 94 times

Subject: Re: error with sort
Posted by David L. Keller on Wed, 12 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> R.Bauer (R.Bauer@fz-juelich.de) writes:
>
>> Dear David,
>>
>> that's all right, but why get I different results on different Operation
>> Systems?
>
> I'm reliably informed that you get different results because
> IDL uses the system supplied qsort() routine on each platform,
> and as they are different implementations, they are free to
> return different sub-orders for the identical elements.
>
> But, as I say, I can't see how it makes much difference,
> although I did appreciate Wayne Landsman's perspective.
>

> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438 E-Mail: davidf@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

Answer: When you have 'parallel' arrays. In my field, you might have pressure, temperature, humidity, wind speed. Say the pressure is nearly steady for much of your data. You sort all of the arrays based on the pressure. You then look at values of temperature, humidity, and wind. For subranges where the pressure has the same value, the other arrays have widely varying values. It is nice to have them come up in the same order for the same values of pressure.

For this to make a 'real' difference, you have to be doing something to the other arrays. For statistical work, I might be putting the arrays into 'bins'. Different ordering, different bins. Harder to debug.

Furthermore, the 'steady' pressure values might be in say chronological order. This arrangement might be meaningful in sampling the data. You might wish to sample 'steady' pressures every 3rd hour. You COULD do this if the pressure stayed in chronological order, would be harder otherwise.

And of course, if you are testing any kind of sorting or sampling or statistical algorithm of your own, a consistent sort is nice. Very annoying to run a statistical algorithm, and get different values with back-to-back runs of the same data, same algorithm.

-- Dave --