Subject: Bizarre slowness from sort()
Posted by Jonathan Joseph on Tue, 23 Apr 2002 14:59:04 GMT
View Forum Message <> Reply to Message

Hello,

My colleague complained of an incredible slowness when trying to sort an array of long integers (on the order of 400,000 of them). I said "you're nuts. Must be a bug in your code" and proceeded to generate a random array of 400,000 long integers and sort them very quickly. "See, it works fine."

So, he showed me his code, and it all looked perfectly normal, and the sort took minutes! The data looked fine (no bizarre values) so we created a save file, opened up a new IDL session tried to sort the data and saw the same slowness!

I've found that the problem occurs on SUN and Windows 2K running IDL 5.5, but not on HP-UX running IDL 5.3. Also, we have found a workaround for the integer case. Adding a small (less than 1) random offset to each element of the array before sorting will make it work quickly and yield the correct result. But this will not work properly unless the array to be sorted is an integer type array, otherwise you could be changing the sort order by adding the random offset. Just converting the array to float or adding a constant offset to each element does NOT fix the problem.

This behavior seems very strange - possibly a bug in IDL. Anyone have any thoughts on this? Can you reproduce this bug on your system?

The save file is located at
http://baritone.astro.cornell.edu/~jj/idl2/
and is called 'sort.bin' (about 1.4 megabytes)

```
IDL> restore, 'sort.bin'
IDL> help, sortme

SORTME          LONG      = Array[376467]

IDL> a = sort(sortme)
```

Works, but takes minutes to return.
If I add a random number between 0 and 0.1 to each element and then sort, it works very rapidly (and produces the correct result since it is not changing the sorting order)

```
IDL> b = sort(sortme + randomu(seed, n_elements(sortme)) * 0.1)
```

Works very fast as expected

Anyone know what's going on?

Thanks.

-Jonathan
