

---

Subject: Re: IDL sorting

Posted by [wlandsman](#) on Thu, 18 Oct 2007 15:50:56 GMT

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

---

On Oct 18, 8:20 am, Wox <nom...@hotmail.com> wrote:

```
> Sorting the resulting 64-bit longs would do the trick, wouldn't it?
>
> function sort,array
> array64=ishft(long64(array),32)+lindgen(n_elements(array))
> return,sort(array64)
> end
```

That is clever. My tests on my V6.4 Linux box find that it is usually faster than `bsort.pro`. It is somewhat slower when there are only a few duplicate values

```
> Btw, I didn't want to ask this, but
> why is IDL's sort doing this?
```

IDL just uses the sort algorithm of the underlying OS. As far as I am aware, the `SORT` function on Linux boxes *does* preserve the order of equal values, but that on Mac and Windows machines does not. I would be interested to hear if anyone finds any exceptions to this rule.

```
> Is there any situation where mixing up
> the order of equal values has a benefit?
```

None that I can think of. But if you just want the fastest `SORT` possible, you might not care what happens to the equal values.

Actually, I think a good suggestion to ITTVIS would be to add a `/preserve_equal` keyword or something similar to `SORT()`. This topic comes up repeatedly.

---