
Subject: Re: matching lists

Posted by [JD Smith](#) on Wed, 06 Apr 2005 23:44:05 GMT

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

On Wed, 06 Apr 2005 20:58:08 +0000, Dick Jackson wrote:

```
> Hi all,
>
> This is a follow-up to a thread from March, 2000, where JD Smith had offered
> functions for returning indices of the values in one array that are found in
> a second array. Perhaps due to a change in what Sort() does with identical
> entries, the "ind_int_SORT" function gave erroneous results.
>
> (I'm on IDL 6.1.1, Windows XP Pro, and I get this:
> IDL> print,sort([1,1])
>      1      0
> )
>
```

I'm on 6.1.1 Linux, and I get:

```
IDL> print,sort([1,1])
      0      1
```

Anybody else get Dick's behavior?

```
> Here's a simple fix that lets it work no matter *what* order Sort() puts the
> indices of identical entries in. (even if it is inconsistent within a single
> call to Sort(!)
>
> =====
> ;; Return the indices of values in a which exists anywhere in b
> ;; (one only for repeated values)
> function ind_int_SORT, a, b
>   flag=[replicate(0b,n_elements(a)),replicate(1b,n_elements(b) )]
>   s=[a,b]
>   srt=sort(s)
>   s=s[srt] & flag=flag[srt]
>   wh=where(s eq shift(s,-1) and flag ne shift(flag, -1),cnt)
>   if cnt eq 0 then return, -1
>   result=srt[wh+flag[wh]]
>   return,result[sort(result)]
> end
> =====
```

Interesting. Although I knew depending on the SORT order was an issue, I didn't think it would actually show up. Your changes make my

SORT based matcher somewhat closer to the MATCH routine written by Don Lindler way back in the 1986, as part of the AstroLib library. In that same thread you mentioned (this thread, I guess!), I had commented:

By the way, I found an implementation I had mentioned a while back on the news group but had forgotten about from the nasa lib called "match" which does pretty much the same thing. It's probably less efficient, since it uses an auxiliary list of indices in addition to the flag vector, instead of just using the sort() results directly, and performs a few more "where" tests as a result. But a similar idea, written first in 1986! Match() is also more immune to changes in sort() than my routine, as a result of carrying around these additional index arrays.

Thanks,

JD
