Subject: Re: Efficient comparison of arrays
Posted by davidf on Mon, 11 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Andy Loughe wrote the other day:

> Given vectors of the type...
>
> a = [1,2,3,4,5]
> b = [3,4,5,6,7]
>
> What is the most efficient way to determine which values that occur in
> a also occur in b (i.e., the values [3,4,5] occur in both a and b).

A friend (wishing anonymity) wrote to me with this solution.
I am not sure how general it is, but it worked with this
test case and several others I made up.

Given a and b:

```
a = [1,2,3,4,5]
b = [3,4,5,6,7]
```

Let,

```
array1 = BYTARR((MAX(a) > MAX(b)) - (MIN(a) < MIN(b)))
array2 = array1
```

Then, let,

```
ind1[a] = 1L
ind2[b] = 1L
```

Finally, let,

```
commonIndex = ind1 * ind2
```

The vector commonIndex now has 1s at the locations where there are
common values in the two sets. In other words,

```
Print, commonIndex
     0   0   0   1   1   1
```

Something similar must be going on in the Venn diagram demo
I found recently in the IDL 5 demos, although a quick look
didn't find the relevant code snippet. Look at d_venn.pro
in the demo source directory.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
Customizable IDL Programming Courses
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com
