Subject: Re: Efficient comparison of arrays
Posted by davidf on Sun, 10 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David R. Klassen writes in response to an Andy Loughe question:

>> 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).
> How about:
> x=where(a eq b)
> This will give you the index numbers in a of those values that are also in b.
> So, to vector of the actual values would be a(x).

Ugh, I don't think so. Maybe it *should* work that way, but it
doesn't. At least not on my computer. :-)

I don't know if this is the most efficient way (it probably isn't),
but this is my off-the-cuff way of solving this problem.

```
FUNCTION A_in_B, a, b
num = N_Elements(a)
vector = FltArr(num)
FOR j=0,num-1 DO BEGIN
  index = Where(a(j) EQ b, count)
  IF count GT 0 THEN vector(j) = 1 ELSE vector(j)=0
ENDFOR
solution = a(Where(vector EQ 1))
RETURN, solution
END
```

When I run the example case above, I get a vector with the values
[3,4,5].

It might be more efficient to sort the arrays and then use some
kind of bubble-sort routine to find the first instance of a in b.
The WHERE function is going to find *all* instances, which is
probably the most inefficient part of this program.

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