Subject: Re: Array intersections
Posted by Andy Loughe on Thu, 26 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

> What is the most efficient way (using IDL, of course) to return
> the index at which two arrays intersect? e.g.
> <snip>

I believe the response of David Fanning does not return the "index"
at which two arrays intersect, but the actual values themselves
(right?).
Here is one solution for what you have asked for...

```
FUNCTION where_array, A, B, IA_IN_B=iA_in_B

; Check for: correct number of parameters.
;                   that A and B have each only 1 dimension
;                   that A and B are defined
if (n_params() ne 2 or (size(A))(0) ne 1 or (size(B))(0) ne 1 $
   or n_elements(A) eq 0 or n_elements(B) eq 0) then begin
   message,'Inproper parameters',/Continue
   message,'Usage: result =
where_array(A,B,[IA_IN_B=ia_in_b]',/Continue
   return,-2
endif

; Parameters exist, let's make sure they are not structures.
if ((size(A))((size(A))(0)+1) eq 8 or $
   (size(B))((size(B))(0)+1) eq 8) then begin
    message,'Inproper parameters',/Continue
    message,'Parameters cannot be of type Structure',/Continue
    return,-2
endif

; Build two matrices to compare.
Na = n_elements(a)
Nb = n_elements(b)
l  = lindgen(Na,Nb)
AA = A(l mod Na)
BB = B(l / Na)

; Compare the two matrices we just created.
I  = where(AA eq BB)
Ia = i mod Na
Ib = i / na

; Normally (without keyword, return index of B that exist in A.
```

```
if keyword_set(iA_in_B) then index = Ia else index = Ib

; Make sure a valid value was found.
if Ia(0) eq -1 or Ib(0) eq -1 then index = -1

return,index
```