I would use Unique to get the non-repeating values. You can then compare
it with the continuous index of A to see if there are repating values.
Then, I'd use Value_locate function to match B against the unique
sub-array of A.
But Histogram would probably be faster.

Pavel

hunter wrote:
>
> Hello,
>
> These seems to be a fairly simple problem but I'm having difficulty coming
> up with an elegant solution.
>
> Let's say I have two vectors of type integer:
>
> A=[0,1,3,3,3,6,7,9,9]
> B=[3,7]
>
> I would like to design a function which returns the indices of all the
> elements of A which appear in B.
>
> i.e.
>
> C=get_match(A,B)
>
> should return
>
> C=[2,3,4,6]
>
> The simplest answer (I believe) is to loop through B and use the where
> command. I just wonder if there is a way to do this without useing the loop,
> as (in reality) the length of B may be very large.
>
> I suppose another possibility is to use the histogram command with
> reverse_indices set. But I think this would still require me to use a loop.
> Although it may be faster since I would only have to call histogram once.
> Any thoughts?
>
> Thanks,
> Eli