
Subject: Re: Array comparison part 2
Posted by [JD Smith](#) on Fri, 04 Oct 2002 20:04:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 03 Oct 2002 09:33:24 -0700, Dick Jackson wrote:

```
> "Sean Raffuse" <sean@me.wustl.edu> wrote in message
> news:anhkqb$ipf$1@newsreader.wustl.edu...
>> Thanks for all the help on my first question. I now have a related
>> question.
>>
>> What is the best (read: fastest) way to do the following:
>>
>> I have an array of coordinates, A = intarr(2,25) and I have another
>> array of a specific location, B = [125,1043]
>>
>> I would like to determine if location B is one of the coordinates in
> A. I
>> need to know if A[*,:] = 125, 1043
>>
>> Is it possible to do this without splitting A?
>
> Oh, sure. Using the "replicate data rather than loop" principle, we
> stretch B to be the same shape as A, then compare. Try this:
>
> nCoords=25
> a=indgen(2,nCoords)
> b=[4,5]
> print,Total(Total(a EQ (Rebin(b, 2, nCoords)), 1) EQ 2) GT 0
>
> (result is 1, there is a match)
>
> b=[4,6]
> print,Total(Total(a EQ (Rebin(b, 2, nCoords)), 1) EQ 2) GT 0
>
> (result is 0, there is no match)
>
> To find *which* one(s) it matches, look at the inner part: Total(a EQ
> (Rebin(b, 2, nCoords)), 1) EQ 2
>
> This will be 1 where 'a' matches a pair of 'b' entries, use Where to
> find which one (or more) it matches.
```

A faster way to do this, if you don't care about the locations where equality occurs, is to use `array_equal()`, which halts as soon as it find an equal value:

```
print, array_equal(a[0,*] ne b[0] OR a[1,*] ne b[1],1b) eq 0b
```

JD
