Subject: Re: Little help on arrays
Posted by mchinand on Mon, 16 Feb 2004 15:12:25 GMT
View Forum Message <> Reply to Message

In article <c0qli8$42a$1@pegasus.fccn.pt>,
Nuno Oliveira <nmoliveira@fc.ul.pt> wrote:
> How do I compare one array with another? I want to avoid comparing
> position per position.
>
> IDL> if [1,1] eq [1,1] then print, 'bingo!'
> % Expression must be a scalar or 1 element array in this context: <BYTE
>   Array[2]>.
> % Execution halted at: $MAIN$
>

If you don't want to compare element by element, you could check to see if the max and min of the difference of the two arrays are the same:

IDL> if max(a-b) eq min(a-b) then print, 'bingo!'

You might want to check to make sure the arrays are the same size first.

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago

Subject: Re: Little help on arrays
Posted by Pepijn Kenter on Mon, 16 Feb 2004 15:12:55 GMT
View Forum Message <> Reply to Message

Nuno Oliveira wrote:
> How do I compare one array with another? I want to avoid comparing
> position per position.
>
> IDL> if [1,1] eq [1,1] then print, 'bingo!'
> % Expression must be a scalar or 1 element array in this context: <BYTE
>   Array[2]>.
> % Execution halted at: $MAIN$
>

You can use the array_equal procedure to compare arrays.

By the way, the result of the expresion [1,1] eq [1,1] is also an array and can therefor not be used in the IF-statement.

```
IDL> print, [1,1] eq [1,1]
   1   1
```

Pepijn.

Subject: Re: Little help on arrays
Posted by Craig Markwardt on Mon, 16 Feb 2004 15:27:52 GMT

View Forum Message <> Reply to Message

Nuno Oliveira <nmoliveira@fc.ul.pt> writes:

```
>  How do I compare one array with another? I want to avoid comparing
>  position per position.
>
>  IDL> if  [1,1] eq [1,1] then print, 'bingo!'
>  % Expression must be a scalar or 1 element array in this context: <BYTE
>     Array[2]>.
>  % Execution halted at: $MAIN$
```

The other posters have good ideas. My own idiom for this comparison
is:

```
  if total(abs(X-Y)) EQ 0 then print, 'bingo!'
```

Craig

Subject: Re: Little help on arrays
Posted by JD Smith on Mon, 16 Feb 2004 19:23:45 GMT

View Forum Message <> Reply to Message

On Mon, 16 Feb 2004 09:27:52 -0600, Craig Markwardt wrote:

```
>
>  Nuno Oliveira <nmoliveira@fc.ul.pt> writes:
>
>
>>  How do I compare one array with another? I want to avoid comparing
>>  position per position.
>>
>>  IDL> if  [1,1] eq [1,1] then print, 'bingo!'
>>  % Expression must be a scalar or 1 element array in this context: <BYTE
>>     Array[2]>.
```

>> % Execution halted at: $MAIN$
>
> The other posters have good ideas. My own idiom for this comparison
> is:
>
>   if total(abs(X-Y)) EQ 0 then print, 'bingo!'

If you have a recent enough IDL, ARRAY_EQUAL() is the way to go, because it stops the comparison as soon as it determines the arrays are not equal, and the second array need not be an array, but can be a scalar. I use this all the time for tricks like:

```
 if ~array_equal(array ge 0,1b)
```

which efficiently determines if any element of array is not ge 0 (i.e. is negative). I also think it's a little more clear what the intention is than TOTAL. Both the TOTAL and ARRAY_EQUAL method are much faster than using WHERE, since they don't have to construct list of indices. Craig's construct is very good when comparing floats, up to some tolerance:

```
 if total(abs(X-Y) ge 1.e-6) eq 0 then print, 'Close enough!'
```

but of course this can also be expressed as:

```
 if array_equal(abs(X-Y) lt 1.e-6,1b) then print,'Close enough!'
```

which would run somewhat faster.

JD

Subject: Re: Little help on arrays
Posted by Foldy Lajos on Mon, 16 Feb 2004 20:13:15 GMT

View Forum Message <> Reply to Message

On Mon, 16 Feb 2004, Craig Markwardt wrote:

>
> Nuno Oliveira <nmoliveira@fc.ul.pt> writes:
>
>
>> How do I compare one array with another? I want to avoid comparing
>> position per position.
>>
>> IDL> if [1,1] eq [1,1] then print, 'bingo!'
>> % Expression must be a scalar or 1 element array in this context: <BYTE
>>   Array[2]>.

>> % Execution halted at: $MAIN$
>
> The other posters have good ideas. My own idiom for this comparison
> is:
>
>   if total(abs(X-Y)) EQ 0 then print, 'bingo!'
>
> Craig
>
>

Hi,

I think

```
    if min(x-y, max=max) eq max then print, 'bingo!' ; for integer arrays
```

is about 20% faster for arrays greater than cache memory.

regards,
lajos

Subject: Re: Little help on arrays
Posted by MKatz843 on Tue, 17 Feb 2004 03:53:52 GMT
View Forum Message <> Reply to Message

mchinand@midway.uchicago.edu (Mike Chinander) wrote in message news:> If you don't want to compare element by element, you could check to see if the max and min of
> the difference of the two arrays are the same:
>
> IDL> if max(a-b) eq min(a-b) then print, 'bingo!'

This gives the wrong result if there's a constant difference between
the arrays. For example.
a = [3,4,5,3]
b = [2,3,4,2]

then
  max(a-b) = max([1,1,1,1]) = 1
yet
  min(a-b) = min([1,1,1,1]) = 1
So they're equal, but the arrays aren't.

M. Katz

Subject: Re: Little help on arrays
Posted by mchinand on Tue, 17 Feb 2004 04:07:44 GMT
View Forum Message <> Reply to Message

In article <4a097d6a.0402161953.238c2a73@posting.google.com>,
M. Katz <MKatz843@onebox.com> wrote:
> mchinand@midway.uchicago.edu (Mike Chinander) wrote in message news:> If
> you don't want to compare element by element, you could check to see if
> the max and min of
>> the difference of the two arrays are the same:
>>
>> IDL> if max(a-b) eq min(a-b) then print, 'bingo!'
>
> This gives the wrong result if there's a constant difference between
> the arrays. For example.
> a = [3,4,5,3]
> b = [2,3,4,2]
>
> then
>   max(a-b) = max([1,1,1,1]) = 1
> yet
>   min(a-b) = min([1,1,1,1]) = 1
> So they're equal, but the arrays aren't.
>
> M. Katz

Yes, I realized that after seeing someone else's similar solution which
does the needed check to make sure the max is equal to zero.

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago