Subject: Re: Speedy way to get compare array elements. . .
Posted by Craig Markwardt on Thu, 14 Nov 2002 04:28:55 GMT
View Forum Message <> Reply to Message

"Sean Raffuse" <sean@me.wustl.edu> writes:
> Exalted newsgroup,
>
> What is the fastest way to compare two adjacent values in an array?
>
> Something like:
>
> whoa_huge_jump = where(A[i+1]-A[i] GT 500)

I usually use this little bit of magic,

whoa_huge_jump = where(A[1:*]-A GT 500)

Now, if you are a goody goody, the correct way to do the subtraction
is A[1:*]-A[0:N-2], but that can get annoying, especially if you have
to figure out N first. The cool thing about the trick is that IDL
automatically truncates the vector A so that it matches the length
of A[1:*].

Some people on the newsgroup thing it is too obscure, but heck, I have
to have something to distinguish my own style.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---

Subject: Re: Speedy way to get compare array elements. . .
Posted by Pavel A. Romashkin on Thu, 14 Nov 2002 16:51:36 GMT
View Forum Message <> Reply to Message

Darn, you definitely *do* have your own style! I think it speaks for
itself when people refer to you by your first name and everyone knows
who they talk about.
Also, if I happen to come across a tiny piece of extremely fast code
that I can't figure out how it works, I immediately think - Craig or JD.
And if there's COMMONs all over, or Heap_Gcs, or at least obscure
objects - that's gotta be David's :-)
Cheers,
Pavel

Craig Markwardt wrote:
>
> Some people on the newsgroup thing it is too obscure, but heck, I have
> to have something to distinguish my own style.
>

Subject: Re: Speedy way to get compare array elements. . .
Posted by the_cacc on Thu, 28 Nov 2002 14:58:34 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:<on4raklqpk.fsf@cow.physics.wisc.edu>...
> ...The cool thing about the trick is that IDL *automatically* truncates
> the vector A so that it matches the length of A[1:*].
>

Ah, but is there any way to not make it do that? Say you have 2
vectors of different lengths:

```
x = findgen(50)
y = findgen(100)
```

and you want to form z = x * y, but have z the same length as y
putting zeros where x has no value. The (sorry) way I am doing it is

```
z = y * 0
z[0:49] = x * y
```

which clearly is too much programmer work since I have to get the
lengths... Any ideas?

Ciao.

Subject: Re: Speedy way to get compare array elements. . .
Posted by marc schellens[1] on Wed, 04 Dec 2002 06:54:18 GMT
View Forum Message <> Reply to Message

trouble wrote:
> Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:<on4raklqpk.fsf@cow.physics.wisc.edu>...
>
>> ...The cool thing about the trick is that IDL *automatically* truncates
>> the vector A so that it matches the length of A[1:*].
>>

>
>
> Ah, but is there any way to not make it do that? Say you have 2
> vectors of different lengths:
>
> x = findgen(50)
> y = findgen(100)
>
> and you want to form z = x * y, but have z the same length as y
> putting zeros where x has no value. The (sorry) way I am doing it is
>
> z = y * 0
> z[0:49] = x * y
>
> which clearly is too much programmer work since I have to get the
> lengths... Any ideas?
>
> Ciao.

I got the impression my news-server is a little slow,
but anyway (as I see no reply so far):

just say:

z[0]=x*y

IDL the copies as many values as there are to z[0] and the
following elements.

marc
