Subject: Re: The IDL way: Find last non-zero value

Posted by Jeremy Bailin on Tue, 25 Aug 2009 02:43:37 GMT

View Forum Message <> Reply to Message

On Aug 24, 8:12 pm, Jeremy Bailin <astroco...@gmail.com> wrote:

```
>  On Aug 24, 4:42 pm, Eric Hudson <ehud...@mit.edu> wrote:
>
>>  Thanks to all of you for the suggestions.  I timed them all on several
>>  of the larger matrices and found that the last suggestion (using
>>  cumulative totals) was about twice as fast as the previous ones, and
>>  about 5 times faster than the brute force loop.  It did depend on how
>>  many zeros ended each line on average.  When the number was small the
>>  loop could win, but for my typical matrices the cumulative total is
>>  the clear winner.
>
>>  Thanks again,
>>  Eric
>
>  I'm shocked that no one's suggested histogram! Here's my take, which
>  I'd wager does at least as well as the cumulative totals:
>
>  sz=size(array,/dimen)
>  ncol=sz[0]
>  nrow=sz[1]
>  evenrows = 2*lindgen(nrow)
>  h = histogram( (array ne 0) + rebin(reform(evenrows,
>  1,nrow),ncol,nrow), min=0, max=2*nrow-1, $
>    reverse_indices=ri)
>  result = ri[ri[evenrows+2]-1] mod ncol
>  w = where(h[evenrows+1] eq 0, nw)
>  if nw gt 0 then result[evenrows[w]]=-1
>
>  -Jeremy.
```

As usual, stupid typo in the last line, which should be:

```
if nw gt 0 then result[w]=-1
```

It looks to me like it doesn't beat the cumulative total version after all, at least on my test case. Oh well! Eric, if you have the timings on them, I'd be very curious to see how they stack up in detail!

-Jeremy.