
Subject: Re: Simple question in IDL, looking for solution, thank you
Posted by [Jeremy Bailin](#) on Tue, 23 Oct 2012 19:57:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 10/23/12 2:07 PM, Heinz Stege wrote:

> On Mon, 22 Oct 2012 14:10:18 -0400, Jeremy Bailin wrote:

>

>> On 10/22/12 7:55 AM, Heinz Stege wrote:

>>> Hi Danxia,

>>>

>>> you didn't ask for a solution without a loop. So here is my simple

>>> answer:

>>>

>>> arr=[5,2,3,1,8,3,1,2,3]

>>> bcg=[1,2,3,2,1,4,2,3,5]

>>> sum=intarr(max(arr)+1)

>>> for i=0,n_elements(bcg)-1 do sum[arr[i]]+=bcg[i]

>>> print,sum[1:*

>>>

>>> Cheers, Heinz

>>>

>>

>> And of course, if you need a very efficient implementation of this (i.e.

>> if your arrays have millions of elements), then read the "chunk

>> indexing" section of JD's HISTOGRAM tutorial

>> http://www.idlcoyote.com/tips/histogram_tutorial.html (you HAVE read

>> JD's HISTOGRAM tutorial, right???)

>>

>> -Jeremy.

>

>

> Hi Jeremy,

>

> I suppose you mean something like the following:

>

> h=histogram(total(bcg,/cumulative,/integer)-1,/binsize,min=0 ,reverse_indices=ri)

> i=ri[0:n_elements(h)-1]-ri[0]

> print,histogram(arr[i],min=1)

>

> The histogram methods in general are very smart. The above code is

> significantly faster than my, which contains the loop. However, from

> my point of view, this is not a good solution.

>

> In case of very many elements within arr (and bcg) and/or big numbers

> within bcg the reverse indices array ri gets very large. The size of

> ri is always greater than total(bcg). IDL may run out of memory.

>

> So I would say, the loop may compete with the reverse indices.

```

>
> When I wrote "simple answer", I had in mind that there must be another
> solution. One without a loop. It is more the "IDL-style". But it is a
> little bit more complex:
>
> ii=sort(arr)
> sarr=arr[ii]
> tot=total(bcg[ii],/cumulative,/integer)
> ;
> ii=where(sarr ne shift(sarr,-1),count)
> if count eq 0 then ii=[n_elements(sarr)-1]
> tot=tot[ii]
> if count ge 2 then tot[1:]=tot
> ;
> sum=lonarr(sarr[n_elements(sarr)-1]+1)
> sum[sarr[ii]]=tot
> ;
> print,sum[1:]
>
> This code has a moderate memory consumption and seems to be a true
> alternative to both, the loop-method and the reverse-indices-method.
>
> A word to the developers of IDL: What about a WEIGHT keyword in the
> histogram function?
>
> print,histogram(arr,weight=bcg,/integer,min=1)
>
> This would be nice. By the way, when I type the line above, IDL
> (Version 8.0.1) says:
>
> % Keyword INTEGER not allowed in call to: HISTOGRAM
> % Error occurred at: $MAIN$
> % Execution halted at: $MAIN$
>
> No integer keyword allowed in the histogram function? Strange! ;-)
>
> Cheers, Heinz
>

```

A couple of notes:

JBIU has a weighted histogram function:

http://astroconst.org/jbiu/jbiu-doc/math/histogram_weight.ht ml

Regarding reverse_indices using lots of memory on sparse histograms: use
VALUE_LOCATE!

http://www.idlcoyote.com/code_tips/valuelocate.html

-Jeremy.
