
Subject: Re: Shortcut needed

Posted by [sclafin](#) on Sun, 16 Nov 1997 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Daniel Williams (williams@idunn.srl.caltech.edu) wrote:

: Hi fellow IDL-users,

: I have a program which runs slower than I like, and wonder if someone
: could help me speed it up a bit.

: My data is an array of structures, call it 'a'. One element of the
: structure, call it 'n' is an integer, another is a quantity which I
: want to add up, call it 'x'.

: There are about 100000 elements in this array. The integer n, marks
: data as being of a certain type. I want to make a new array, call it
: 'y', which has as its nth element the sum of all the x's which have a
: n-value of n. I currently do this as follows,

```
: for n=0, n_max do begin  
:   y[n] = total( a[where(a.n eq n)].x )  
: endfor
```

: The trouble is that the loop is slow, as for loops always are. Does
: anyone out there know a better way to do this kind of sorting?

Some weeks ago the virtues of the histogram function for sorting
problems were discussed in this news group. I haven't tested the
the following solution, but if you have a really slow process,
it might be worth a try. There is still a "for" loop, but the
"where" function does not have to be called each time through the loop.

```
h = histogram(a.n, Min=0, Max=n_max, Reverse_Indices=r)  
h = 0b  
y = intarr(n_max + 1)
```

```
for i=0,n_max do begin  
  if (r[i+1] gt r[i]) then begin  
    y[i] = total(a[r[r[i]:r[i+1]-1]].x)  
  endif  
endfor
```