
Subject: Re: Ordered index array

Posted by [Paolo Grigis](#) on Thu, 08 Sep 2005 08:22:30 GMT

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

David Fanning wrote:

> David Fanning writes:

>

>

>> How about this:

>>

>> PRO TEST

>> a = [3,6,2,1,2,7,1,1]

>> h = Histogram(a, Reverse_Indices=ri, Min=0)

>> b = Indgen(N_Elements(h)) + 1

>> c = Intarr(N_Elements(h))

>> FOR j=0,N_Elements(h)-1 DO BEGIN

>> IF ri[j+1] NE ri[j] THEN \$

>> c[ri[ri[j]:ri[j+1]-1]] = Min(b[ri[ri[j]:ri[j+1]-1]])

>> ENDFOR

>> Print, c

>> END

>>

>> 1 2 3 4 3 6 4 4

>>

>> Note that your original example is wrong. :-)

>

>

> Whoops! *I* didn't use the original example either (although

> yours is still wrong!).

>

> Here is a more complete solution, using the original data:

>

> PRO TEST

> a = [3,6,2,1,2,8,1,1]

> h = Histogram(a, Reverse_Indices=ri, Min=0)

> b = Indgen(N_Elements(h)) + 1

> c = Intarr(N_Elements(h))

> FOR j=0,N_Elements(h)-1 DO BEGIN

> IF ri[j+1] NE ri[j] THEN \$

> c[ri[ri[j]:ri[j+1]-1]] = Min(b[ri[ri[j]:ri[j+1]-1]])

> ENDFOR

> Print, c[0:N_Elements(a)-1]

> END

But this will fail if 'a' has more elements than the number of
its different values, for instance a=[3,3,1,2,3,2,2,1,2,3].

One could try this:

PRO test

```
a=[3,3,1,2,3,2,2,1,2,3]
```

```
b=a
```

```
c=intarr(n_elements(a))
```

```
h=histogram(a,min=1,reverse_ind=ri)
```

```
done=0
```

```
rank=1
```

```
WHILE NOT done DO BEGIN
```

```
  actual_value=b[0]
```

```
  ind_actual_value=ri[ri[actual_value-1]:ri[actual_value]-1]
```

```
  c[ind_actual_value]=rank
```

```
  rank=rank+1
```

```
  indremove=where(b NE actual_value,count)
```

```
  IF count GT 0 THEN b=b[indremove] ELSE done=1
```

```
ENDWHILE
```

```
print,a
```

```
print,c
```

```
END
```

but it will get inefficient as the numbers of different values in 'a' grows, as the code in the loop get called more and more times...

Ciao,

Paolo

>

> Cheers,

>

> David

>

>