Subject: Re: Array sorting by row
Posted by humphreymurray on Fri, 18 Aug 2006 02:52:10 GMT
View Forum Message <> Reply to Message

Cheers Jean, that code of yours runs well.

However, I decided to switch back to my original code with the for loop. This was because the time taken to sort increases significantly with the number of elements to be sorted. In your approach, you are sorting the entire array, and then working out what elements are in each row.

My original code (with the loops) runs faster (for me anyway), because each sort operation is sorting a smaller subset of the numners, and so is quicker.

That is, unless somebody can proof me wrong :-p

Humphrey

Jean H. wrote:

```
>  a = [[4,2,0,5],[9,0,1,5],[0,4,2,1],[1,2,3,4]]
>  b = sort(a) ;sort the whole array
>
>  sizeX = 4 ;(use DIM instead)
>  sizeY = 4
>  nbElements = n_elements(a)
>
>  c = b/sizeX  ;tells you, for each element of b, which line of A the
>      ;index correspond to
>
>  useless = histogram(c, reverse_indices = ri)
>  ;get the reverse indices of the histogram. It tells you, for each line
>  of a (print useless, you will get 4 4 4 4), the indices of b that
>  correspond to it.
>
>  ;get the sorted indices, resorted by lines. You have, for sure, 4 bin
>  here, so the first indice of b is located at ri[5].
>  d = b[ri[sizeX+1:sizeX+nbElements]]
>
>  minus = indgen(sizeY) * sizeX
>  minusBIG = transpose(rebin(minus,sizeX,sizeY))
>  ;you want the indices on each line and not on the whole array (if that's
>  what you want, d is fine for you then)
>
>  result =  reform(d-minusBIG, sizeX,sizeY)
>
```

```
>
> IDL> print, result
>           2          1          0          3
>           1          2          3          0
>           0          3          2          1
>           0          1          2          3
>
>
> I suggest you to learn to use the histogram... it's wonderful what you
> can do with it!
>
> Jean
>
>
> humphreymurray@gmail.com wrote:
>> Hi,
>>
>> In IDL, is there a way to independly sort the columns of a 2d matrix
>> without looping through and sorting each row individually?
>>
>> Currenly I'm using:
>>
>> for x = long(0), x_size - 1 do begin
>>    sorted_indexs[0,x] = sort(matrix[*,x])
>> endfor
>>
>> For example, if the matrix contained the following values:
>>
>> 4 2 0 5
>> 9 0 1 5
>> 0 4 2 1
>> 1 2 3 4
>>
>> I want the result matrix to contain index's like:
>>
>> 2 1 0 3
>> 1 2 3 0
>> 0 3 2 1
>> 0 1 2 3
>>
>> Here, each column is sorted as if it's an independant vector.
>>
>> Cheers.
>>
```