Subject: Re: remove duplicate elements from a multi-dimensional array efficiently in IDL
Posted by Jeremy Bailin on Fri, 01 May 2009 18:36:18 GMT
View Forum Message <> Reply to Message

On May 1, 1:47 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
> On May 1, 12:13 pm, "chenb...@gmail.com" <chenb...@gmail.com> wrote:
>
>
>
>> Hello, everyone!
>
>> Is there anyone knows a routine in IDL that be capable to remove
>> duplicate elements from a multi-dimensional array efficiently? I 'm
>> now working with huge arrays, and I have written one by myself, it
>> works but is with low efficiency.
>
>> example of my problem:
>> the input array:
>> 1,10,9,100,200
>> 2,11,8,101,201
>> 2,11,8,101,201
>> 3,10,9,100,200
>> 4,7,12,99,199
>> 2,11,8,101,201
>
>> goal:
>> remove the duplicate elements with the same values for the second and
>> the third column.
>
>> expected output:
>> 1,10,9,100,200
>> 2,11,8,101,201
>> 4,7,12,99,199
>
>> Thanks for your help!
>
>> Bo
>
> How's this:
>
> input = [[1,10,9,100,200],[2,11,8,101,201],[2,11,8,101,201],$
> [3,10,9,100,200],[4,7,12,99,199],[2,11,8,101,201]]
>
> ; Step 1: Map your columns 2 and 3 into a single unique index
> (requires ORD from JBIU):
> col2ord = ord(input[1,*])
> col3ord = ord(input[2,*])

```
>  index = col2ord + (max(col2ord)+1)*col3ord
>
>  ; Step 2: Use histogram to find which ones have the same unique index
>  h = histogram(index, reverse_indices=ri)
>
>  ; Step 3: Get the first one in each bin, and put back in sorted order
>  keep = ri[ri[where(h gt 0)]]
>  keep = keep[sort(keep)]
>
>  ; Step 4: Print them out:
>  print, input[*,keep]
>
>        1     10      9    100    200
>        2     11      8    101    201
>        4      7     12     99    199
>
>  -Jeremy.
```

Incidentally, if you're dealing with huge arrays and run into memory problems with the histogram, you can replace:

index = col2ord + (max(col2ord)+1)*col3ord

with

index = ord(col2ord + (max(col2ord)+1)*col3ord)

which will make the histogram as compact as possible.

-Jeremy.
