Subject: Re: remove duplicates WITHOUT sorting
Posted by Vince Hradil on Wed, 14 Feb 2007 15:32:45 GMT
View Forum Message <> Reply to Message

```
On Feb 13, 9:18 am, "rpert...@gmail.com" <rpert...@gmail.com> wrote:
>  On Feb 12, 4:15 pm, JD Smith <jdsm...@as.arizona.edu> wrote:
>
>
>
>>  On Mon, 12 Feb 2007 09:04:45 -0800, rpert...@gmail.com wrote:
>>>  Hello All,
>>>  I have an array of 2xn which is a list of coordinates (x,y positions)
>>>  and I need to remove any duplicates in this array. Since they are x,y
>>>  coordinates I cannot really sort the array, hence cannot use the Uniq
>>>  function. Is there another way of doing this?
>
>>  It's easiest to recast as 1D.  HIST_2D (or HIST_ND) does this for you, but
>>  it's easy to do yourself:
>
>>   index=x + (max(x)+1)*y
>
>>  and then using UNIQ on this list of indices should give you the row
>>  positions of unique coordinates.  HISTOGRAM can work as well (either
>>  with HIST_ND, or by first constructing this index vector above), and
>>  it will be faster, but, as usual, will consume lots of memory (and
>>  potentially be very slow) if your coordinates are sparsely sprinkled over
>>  a large range of values (your current example is somewhat sparse, but not
>>  horrible).  UNIQ, with its SORT based implementation, doesn't suffer from
>>  that issue.
>
>>  JD
>
>  Thank You All for responding.
>  I basically did sort in the end, but sorted the x coords and
>  corresponding y coords stayed with
>  its pair. Then did a shift to eliminate recurrences:
>
>  SortIndex=Sort(AllCoords[0,*])
>  for j=0,1 Do AllCoords[j,*] = AllCoords[j,sortindex]
>  Print,"Sorted All Cords is",AllCoords
>  print,' '
>
>  B=AllCoords - Shift(AllCoords,2)
>  print,' '
>  C=AllCoords[*,where(B[0,*] ge 0.5 or B[1,*] ge 0.6)]
>  csize=size(C,/dimensions)
>  print,'size c' ,csize[0],csize[1]
```

So did you try the cast-to-complex, recast-to-float approach?