Subject: Re: setintersection assumes sets have no repetitions?
Posted by Jeremy Bailin on Sat, 23 Feb 2013 21:15:10 GMT
View Forum Message <> Reply to Message

On 2/23/13 2:46 PM, Paulo Penteado wrote:
>
> I am using Coyote's set functions, and I noticed that when I use some
> (not any) sets with repetitions as input, it crashes:
>
> IDL>
> a=[1,2,3,4,5]
> IDL>
> b=[1,2,2,2]
> IDL>
> print,setintersection(a,b,indices_a=ia,indices_b=ib,position s=pos)
> % Compiled module: SETINTERSECTION.
> % Compiled module: REVERSEINDICES.
> % Compiled module: ERROR_MESSAGE.
>
> Traceback Report from SETINTERSECTION:
>
> % Out of range subscript encountered: BINDICES.
> % Execution halted at: SETINTERSECTION 192 /software/idl/
> others/idl-coyote-read-only/coyote/setintersection.pro
> % $MAIN$
> -1
>
> Is this the expected behavior? Are the input sets supposed not to have
> repetitions? The documentation suggests they may have repeated
> elements, thus making the positions array different from indices_a.
>

By definition, a set only has one copy of each element, so I wouldn't be
surprised if things fail if it's not a true set.

-Jeremy.

Subject: Re: setintersection assumes sets have no repetitions?
Posted by David Fanning on Sun, 24 Feb 2013 06:23:03 GMT
View Forum Message <> Reply to Message

Paulo Penteado writes:

> I am using Coyote's set functions, and I noticed that when I use some
> (not any) sets with repetitions as input, it crashes:
>
> IDL>

> a=[1,2,3,4,5]
> IDL>
> b=[1,2,2,2]
> IDL>
> print,setintersection(a,b,indices_a=ia,indices_b=ib,position s=pos)
> % Compiled module: SETINTERSECTION.
> % Compiled module: REVERSEINDICES.
> % Compiled module: ERROR_MESSAGE.
>
> Traceback Report from SETINTERSECTION:
>
> % Out of range subscript encountered: BINDICES.
> % Execution halted at: SETINTERSECTION 192 /software/idl/
> others/idl-coyote-read-only/coyote/setintersection.pro
> % $MAIN$
> -1
>
> Is this the expected behavior? Are the input sets supposed not to have
> repetitions? The documentation suggests they may have repeated
> elements, thus making the positions array different from indices_a.

Well, I don't know. That code was added at the request of Mr. Stockwell. Let's see if he has any ideas about this. :-)

For the moment, I would consider commenting that section of the code out.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: setintersection assumes sets have no repetitions?
Posted by penteado on Tue, 05 Mar 2013 18:44:04 GMT
View Forum Message <> Reply to Message

I just noticed that the documentation changed, to say that indices_a and indices_b require the intersection elements to be unique, while positions does not.

I understand the reasons to assume the sets have unique elements (both from the mathematical concept and from practical applications like identifiers from databases). But since the positions keyword was there to handle repetitions, I got confused when the code crashed on arrays with repetitions.

The use I had with repeated elements happened when I was working with matches of my list of sources in the sky (each with its unique identifier) with other such lists (with their own unique identifiers), by finding which objects in one list were within a certain distance of each object in the other list. Occasionally, one object in my list would have more than one match, and, as such, its identifier would appear more than once in the list of results.

I eventually resorted to using histogram, through my wrapper that puts the reverse indices into hashes or lists:

```
a=[1,2,3,4,5]
b=[1,2,2,2]
h=histogram_pp(a,binsize=1,reverse_hash=rh)
print,rh
;5:           4
;1:           0
;3:           2
;2:           1
;4:           3
ib=lonarr(n_elements(b))
foreach el,b,i do ib[i]=rh[el]
print,ib
;           0           1           1           1
print,b[ib]
;       1       2       2       2
```

histogram_pp is at
http://www.ppenteado.net/idl/pp_lib/doc/histogram_pp.html