Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by David Fanning on Wed, 17 Oct 2001 03:01:36 GMT
View Forum Message <> Reply to Message

Ted Cary (tedcary@yahoo.com) writes:

> Is there any programming technique for finding the intersection of two sets
> (arrays) of numbers without using WHERE in a loop to search the larger array
> for every element in the smaller array? It seems like a very clumsy way to
> find values shared by both arrays, especially with integer sets/arrays.

How about the very tiny program, SetIntersection,
which uses--what else--a Histogram! :-)

http://www.dfanning.com/tips/set_operations.html

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by Mark Hadfield on Wed, 17 Oct 2001 03:37:40 GMT
View Forum Message <> Reply to Message

From: "David Fanning" <david@dfanning.com>
> http://www.dfanning.com/tips/set_operations.html

Well b****r me, isn't that clever!

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--
Posted from clam.niwa.cri.nz [202.36.29.1]
via Mailgate.ORG Server - http://www.Mailgate.ORG

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by David Fanning on Wed, 17 Oct 2001 03:52:23 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> Well b****r me, isn't that clever!

It is, isn't it. I've had occasion to make
extensive use of these little functions lately,
and I have to admit they work like a charm IF
you pass them the correct arguments. In practice,
I've had to add a bit of error checking to them. :-)

Cheers,

David

P.S. Let's just say an output COUNT keyword is
probably a good idea.

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by John-David T. Smith on Wed, 17 Oct 2001 05:00:34 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Ted Cary (tedcary@yahoo.com) writes:
>
>> Is there any programming technique for finding the intersection of two sets
>> (arrays) of numbers without using WHERE in a loop to search the larger array
>> for every element in the smaller array? It seems like a very clumsy way to
>> find values shared by both arrays, especially with integer sets/arrays.
>
> How about the very tiny program, SetIntersection,
> which uses--what else--a Histogram! :-)
>
> http://www.dfanning.com/tips/set_operations.html

It's amazing how much recycled information flows through the newsgroup, if you watch it long enough. I remember just like it was 4 years ago the detailed discussions with which we whiled away our days, concerning value-based intersection vs index-based intersection, order N vs. unknown order operations, etc.

I do, however, still cringe when I read on that page of yours:

"These routines are faster than previously published functions, e.g. Contain..."

since I whipped up contain() to prevent the problem of an operation scaling out of control on sparse data sets (as set_intersection does -- a feature they warn you of). You can read all about a variety of intersection methods if you like:

http://groups.google.com/groups?selm=38CBF8B6.5BF0AB50%40ast ro.cornell.edu

In a classic example posed by Mark Fardal, you are matching up social security numbers in two lists containing age and income. The set_intersection style solution fails miserably here, and to a lesser degree for any arrays which are somewhat sparse (where *somewhat* seems to be about 1 in 10, depending on lots of factors).

Ah, the glory days.

JD

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by R.Bauer on Wed, 17 Oct 2001 06:58:51 GMT
View Forum Message <> Reply to Message

Ted Cary wrote:
>
> Is there any programming technique for finding the intersection of two sets
> (arrays) of numbers without using WHERE in a loop to search the larger array
> for every element in the smaller array? It seems like a very clumsy way to
> find values shared by both arrays, especially with integer sets/arrays.
>

> Thanks.

Dear Ted,
we have some routines which might be useful for you too.

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/a_and_b.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/a_or_b.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/a_not_b.tar.gz

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/a_xor_b.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/veklogic.tar.gz

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/indexlogic.tar.gz

For licensing of more routines please have a look at
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

Reimar
--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by David Fanning on Wed, 17 Oct 2001 13:58:53 GMT
View Forum Message <> Reply to Message

John-David Smith (jdsmith@astro.cornell.edu) writes:

> It's amazing how much recycled information flows through the newsgroup, if you
> watch it long enough. I remember just like it was 4 years ago the detailed
> discussions with which we whiled away our days, concerning value-based
> intersection vs index-based intersection, order N vs. unknown order operations,
> etc.
>
> I do, however, still cringe when I read on that page of yours:
>
> "These routines are faster than previously published functions, e.g. Contain..."
>
> since I whipped up contain() to prevent the problem of an operation scaling out
> of control on sparse data sets (as set_intersection does -- a feature they warn
> you of). You can read all about a variety of intersection methods if you like:

>
> http://groups.google.com/groups?selm=38CBF8B6.5BF0AB50%40ast ro.cornell.edu
>
> In a classic example posed by Mark Fardal, you are matching up social security
> numbers in two lists containing age and income. The set_intersection style
> solution fails miserably here, and to a lesser degree for any arrays which are
> somewhat sparse (where *somewhat* seems to be about 1 in 10, depending on lots
> of factors).
>
> Ah, the glory days.

Alright, alright. I'm spending the day updating my web
page. :-(

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by Craig Markwardt on Wed, 17 Oct 2001 15:05:25 GMT
View Forum Message <> Reply to Message

John-David Smith <jdsmith@astro.cornell.edu> writes:

> David Fanning wrote:
>>
>> Ted Cary (tedcary@yahoo.com) writes:
>>
>>> Is there any programming technique for finding the intersection of two sets
>>> (arrays) of numbers without using WHERE in a loop to search the larger array
>>> for every element in the smaller array? It seems like a very clumsy way to
>>> find values shared by both arrays, especially with integer sets/arrays.
>>
>> How about the very tiny program, SetIntersection,
>> which uses--what else--a Histogram! :-)
>>
>>   http://www.dfanning.com/tips/set_operations.html
>
>
> It's amazing how much recycled information flows through the newsgroup, if you

> watch it long enough. I remember just like it was 4 years ago the detailed
> discussions with which we whiled away our days, concerning value-based
> intersection vs index-based intersection, order N vs. unknown order operations,
> etc.
>
...
> In a classic example posed by Mark Fardal, you are matching up social security
> numbers in two lists containing age and income. The set_intersection style
> solution fails miserably here, and to a lesser degree for any arrays which are
> somewhat sparse (where *somewhat* seems to be about 1 in 10, depending on lots
> of factors).

Hi JD--

Thanks for beating me to the punch. The HISTOGRAM method is indeed very cool for a new learner, but it definitely starts to suck air (and memory) when the data sets become sparse.

Long ago (1 year?) I tried to collect all the various algorithms that were being discussed, and some that weren't yet, to do set operations. CMSET_OP has the dreaded "CM" prefix, but it also knows how to do intersections, unions, and exclusive or's. It can do X AND NOT Y type intersections as well, in one self contained function.

The syntax is:

```
x_and_y = cmset_op(X, 'AND', y)
```

It can return by value or index.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by John-David T. Smith on Wed, 17 Oct 2001 16:51:24 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
>

> John-David Smith <jdsmith@astro.cornell.edu> writes:
>
>> David Fanning wrote:
>>>
>>> Ted Cary (tedcary@yahoo.com) writes:
>>>
>>>> Is there any programming technique for finding the intersection of two sets
>>>> (arrays) of numbers without using WHERE in a loop to search the larger array
>>>> for every element in the smaller array? It seems like a very clumsy way to
>>>> find values shared by both arrays, especially with integer sets/arrays.
>>>
>>> How about the very tiny program, SetIntersection,
>>> which uses--what else--a Histogram! :-)
>>>
>>> http://www.dfanning.com/tips/set_operations.html
>>
>>
>> It's amazing how much recycled information flows through the newsgroup, if you
>> watch it long enough. I remember just like it was 4 years ago the detailed
>> discussions with which we whiled away our days, concerning value-based
>> intersection vs index-based intersection, order N vs. unknown order operations,
>> etc.
>>
> ...
>> In a classic example posed by Mark Fardal, you are matching up social security
>> numbers in two lists containing age and income. The set_intersection style
>> solution fails miserably here, and to a lesser degree for any arrays which are
>> somewhat sparse (where *somewhat* seems to be about 1 in 10, depending on lots
>> of factors).
>
> Hi JD--
>
> Thanks for beating me to the punch. The HISTOGRAM method is indeed
> very cool for a new learner, but it definitely starts to suck air (and
> memory) when the data sets become sparse.
>
> Long ago (1 year?) I tried to collect all the various algorithms that
> were being discussed, and some that weren't yet, to do set operations.
> CMSET_OP has the dreaded "CM" prefix, but it also knows how to do
> intersections, unions, and exclusive or's. It can do X AND NOT Y type
> intersections as well, in one self contained function.
>
> The syntax is:
>
> x_and_y = cmset_op(X, 'AND', y)
>
> It can return by value or index.

Ahah, a nice update since last I looked. I'm sure the exact break between histogram vs. sort is machine dependent, but your defaults seem logical.

There's one more thing I should point out in support of the much maligned ARRAY method, as exemplified in the where_array() routine originally by Dan Carr at RSI: it works for *any* IDL type.

In as much as comparisons like:

```
a=ptr_new('test') & b=a
print, b eq a
```

and

```
a=obj_new('myClass') & b=a
print, b eq a
```

work, you can do intersections on lists of pointers, lists of objects, etc., by using the array method. The underlying IDL operation which is data-type agnostic is simply array indexing, so in the context of the REFORM/REBIN tutorial, you can use the awkward "lindgen(n,m) mod m"-type method (of which where_array is a special case) to perform flexible operations on any type of array. Just beware of the N^2 performance.

I'm also not sure how sort is defined on pointer and object arrays... probably by heap variable number, in which case that one should work too.

JD

Subject: Re: Intersection of 2 sets--Beginner IDL question
Posted by Craig Markwardt on Wed, 17 Oct 2001 21:49:02 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@astro.cornell.edu> writes:
> Craig Markwardt wrote:
...
>> Long ago (1 year?) I tried to collect all the various algorithms that
>> were being discussed, and some that weren't yet, to do set operations.
>> CMSET_OP has the dreaded "CM" prefix, but it also knows how to do
>> intersections, unions, and exclusive or's. It can do X AND NOT Y type
>> intersections as well, in one self contained function.
>>
>> The syntax is:
>>
>> x_and_y = cmset_op(X, 'AND', y)

>>
>> It can return by value or index.
>
> Ahah, a nice update since last I looked. I'm sure the exact break
> between histogram vs. sort is machine dependent, but your defaults seem
> logical.
>
> There's one more thing I should point out in support of the much
> maligned ARRAY method, as exemplified in the where_array() routine
> originally by Dan Carr at RSI: it works for *any* IDL type.
>
> In as much as comparisons like:
>
> a=ptr_new('test') & b=a
> print, b eq a
>
> and
>
> a=obj_new('myClass') & b=a
> print, b eq a
>
> work, you can do intersections on lists of pointers, lists of objects,
> etc., by using the array method. The underlying IDL operation which is
> data-type agnostic is simply array indexing, so in the context of the
> REFORM/REBIN tutorial, you can use the awkward "lindgen(n,m) mod m"-type
> method (of which where_array is a special case) to perform flexible
> operations on any type of array. Just beware of the N^2 performance.
>
> I'm also not sure how sort is defined on pointer and object arrays...
> probably by heap variable number, in which case that one should work
> too.

Hi JD--

That's a really good point, and worth exploring further. In principle
it could be as easy as just loosening the restriction on the data
type, but I guess I was just being conservative.

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------