

---

Subject: Re: Efficient comparison of arrays  
Posted by [J.D. Smith](#) on Tue, 12 Aug 1997 07:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

```
>
> Augh, it's too late for this:
>
> I wrote this:
>
>> Given a and b:
>>
>>   a = [1,2,3,4,5]
>>   b = [3,4,5,6,7]
>>
>> Let,
>>
>>   array1 = BYTARR((MAX(a) > MAX(b)) - (MIN(a) < MIN(b)))
>>   array2 = array1
>>
>> Then, let,
>>
>>   ind1[a] = 1L
>>   ind2[b] = 1L
>>
>> Finally, let,
>>
>>   commonIndex = ind1 * ind2
>>
>> The vector commonIndex now has 1s at the locations where there are
>> common values in the two sets. In other words,
>>
>>   Print, commonIndex
>>       0 0 0 1 1 1
>
> When I meant to write this:
>
> Given a and b:
>
>   a = [1,2,3,4,5]
>   b = [3,4,5,6,7]
>
> Let,
>
>   array1 = BYTARR((MAX(a) > MAX(b)) - (MIN(a) < MIN(b)))
>   array2 = array1
>
> Then, let,
```

```

>
> array1[a] = 1L
> array2[b] = 1L
>
> Finally, let,
>
> commonIndex = array1 * array2
>
> The vector commonIndex now has 1s at the locations where there are
> common values in the two sets. In other words,
>
> Print, commonIndex
>    0  0  0  1  1  1
>
> --

```

There is an error in this code. Alex Schuster presents a similar solution, but without the error. The problem is you should be subtracting ( $\min(a) < \min(b)$ ) from a and b as such:

```
array1[a- (min(a) < min(b))]=1L
```

and then add the minimum of the two vectors to the location in the commonIndex vector to get the final common values.

Otherwise, there will not, in general, be enough room in the index arrays to mark all the data values. It is just an accident that it works for [1,2,3,4,5],[3,4,5,6,7] ... try [1,2,3,4,5,6,7],[3,4,5,6,7,8] and you'll see the problem.

Another question with the process... what happens when you don't have a well grouped set of integers... e.g [1,2,3,4,5] and [3,4,10000,900,2] ... lots of wasted zeroes in those index arrays to determine this one.

JD

---