Subject: Re: Efficient comparison of arrays
Posted by John Votaw on Wed, 20 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I very frequently have a task similar to finding common elements in arrays as discussed in this thread. I have a large array A and an array B which is known to be a subset of A. No elements in A or B are repeated. This situation occurs when A is an array of indicies into an image volume forming a region of interest and B are the indicies of some feature you would like to remove from the region of interest. The problem is to return an array that contains the elements of A that are not in B.

Following the lead of J. D. Smith, I wrote the following routine:

```
function eliminate,a,b
  c =[a,b]
  cs=c(sort(c))
  keepers=where(cs ne shift(cs,1) and cs ne shift(cs,-1), count)
  if count ne 0 then return,cs(keepers)
  return,-1
end
```

The brute force method:

```
function eliminate_bf,a,b
  mn=min(a)
  c=[mn-2,a,mn-1]  ;remove possibility of end effects
  for i=0,n_elements(b)-1 do begin
    j=(where(b(i) eq c))(0)
    c=[c(0:j-1),c(j+1:*)]
  endfor
  return,c(1:n_elements(c)-2)
end
```

In my applications, a has about 20000 elements and b has between 1 and 1000. If the number of elements in b is less than 35, then the brute force method is faster, otherwise eliminate is faster -- very much so. When the number of elements in b is 100, it is 3 times faster.

Does anyone have another algorithm or comments?

John R. Votaw
votaw@commander.eushc.org