Subject: Re: Searching in a sorted array
Posted by Martin Schultz on Mon, 26 Apr 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Tri VU KHAC wrote:
>
> Hi folks,
>
> subject : Searching an element replying to a criteria in an unique &
> sorted array
>
> I think every programmer can make an efficient Finder function replying
> to this request.
> For example: Looking for the last element smaller than or equal 100 in a
> sorted array.
>
> If you do this program in C/C++, you must use the recursive function or
> repeat, for, while.
> My question is, does IDL have a built-in function for doing this.
> Why this ? Well, because of computing time.
>
> Thank you for your suggestion.
>
> Best regards,
>
> Tri.

Please fidn attached a little binary search routine that I hacked a few
months ago for another request on this group. Hope, it helps,
Martin.

--
 -- -------
Dr. Martin Schultz
Department for Engineering&Applied Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
 -- -------

```
; $Id: search.pro,v 1.10 1999/01/22 20:12:17 mgs Stab $
 ;---------------------------------------------------------- --
;+
; NAME:
;       SEARCH (function)
```

```
;
; PURPOSE:
;       Perform a binary search for the data point closest
;       to a given value. Data must be sorted.
;
; CATEGORY:
;       Math
;
; CALLING SEQUENCE:
;       index = SEARCH( DATA, VALUE )
;
; INPUTS:
;       DATA -> a sorted data vector
;
;       VALUE -> the value to look for
;
; KEYWORD PARAMETERS:
;       none.
;
; OUTPUTS:
;       The function returns the index of the nearest data
;       point.
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;       This routine is much faster than WHERE or MIN for
;       large arrays. It was written in response to a newsgroup
;       request by K.P. Bowman.
;
; EXAMPLE:
;       test = findgen(10000)
;       print,search(test,532.3)
;       ; prints 532
;
; MODIFICATION HISTORY:
;       mgs, 21 Sep 1998: VERSION 1.00
;
;-
; Copyright (C) 1998, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine search"
 ;----------------------------------------------------------- --


function search,data,value


   ; search first occurence of value in data set
   ; data must be sorted

   ; simple error checking on data and value
   if (n_elements(value) eq 0) then begin
      message,'Must supply sorted data array and value),/CONT
      return
   endif

   ndat = n_elements(data)

   try = fix(0.5*ndat)
   step = 0.5*try

   ; find index of nearest points
   while (step gt 1) do begin
      if (data[try] gt value) then $
         try = try-step $
      else   $
         try = try+step
      step = fix(0.5*(step+1))
   endwhile

   ; now get the data point closest to value
   ; can only be one out of three (try-1, try, try+1)
   dummy = min( abs(value-data[try-1:try+1]), location )

   return,try+location-1

end
```

File Attachments

1) search.pro, downloaded 73 times