
Subject: Re: Search routines

Posted by [Martin Schultz](#) on Mon, 21 Sep 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman wrote:

>

>> Kenneth P. Bowman wrote:

>>> IDL has a pretty good SORT routine, but no SEARCH routine [...]

Well, how about the one attached? I tested it with a 10000 element array, and it is in fact about a factor of ten faster than MIN or WHERE.

Enjoy,
Martin.

Dr. Martin Schultz
Department for Earth&Planetary 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/>

;-----
;+
; 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 occurrence 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 115 times
-