
Subject: Re: Looking for a search routine

Posted by on Fri, 24 Nov 1995 08:00:00 GMT

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

jvkepner@airy.Princeton.EDU (Jeremy Kepner) wrote:

> I have two 1D floating point vectors X and Y.
> Y contains values sorted in increasing order. I am
> looking for a function that for each element in the
> X array, X(i), will return the index of the element in the
> Y array that is nearest to X(i). Ideally it would
> be a function that would look something like
>
> y_ids = SEARCH(X,Y)
>
> -Jeremy Kepner
> Dept. of Astrophysics
> Princeton University

You'll probably have to do it element by element.
Maybe the quickest algo. would be a binary search in
the y array. Still, a quick and dirty approach would be
something like:

```
FUNCTION Search, x, y
  n_elem = N_ELEMENTS(x)
  ndx = lonarr(n_elem)

  FOR i = 0, n_elem - 1 DO BEGIN
    diff = abs(y - x(i))
    ndx(i) = (where (diff EQ min(diff)))(0)
  ENDFOR

  RETURN, ndx
END
```

This function does not require Y to be sorted.

Better suggestions, IDL hackers ?

--
/* Frank J. I¹/₂ynes | frank@spacetec.no /*
/* Spacetec a.s | Phone: +47 77684500 Fax: +47 77655859 /*
/* Prestvannv. 38, | /*
/* N-9005 Tromsø¹/₂, Norway | (...with the bravery of being out of range!) /*

Subject: Re: Looking for a search routine
Posted by [agrap](#)s on Sat, 25 Nov 1995 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

jvkepner@airy.Princeton.EDU (Jeremy Kepner) writes:

> I have two 1D floating point vectors X and Y.
> Y contains values sorted in increasing order. I am
> looking for a function that for each element in the
> X array, X(i), will return the index of the element in the
> Y array that is nearest to X(i). Ideally it would
> be a function that would look something like

> y_ids = SEARCH(X,Y)

Here's a routine that I've used successively many times that does mostly what you want. It uses a bisection algorithm and is reasonably fast. I don't know all of the routine's origins (I know that an atmospheric group at GSFC had something to do with it) and the code isn't pretty, but it works.

Amara

```
function indx,x,t
;+
; NAME:
;   indx
; PURPOSE:
;   finds the index into vector x whose element is closest to the value t.
; CATEGORY:
;   array math
; CALLING SEQUENCE:
;   z = indx(x,t)
; INPUTS:
;   x = vector of function values
;   t = function value for which we want to find the nearest index
; OPTIONAL INPUT PARAMETERS:
; KEYWORD PARAMETERS:
; OUTPUTS:
;   returns the index
; OPTIONAL OUTPUT PARAMETERS:
; COMMON BLOCKS:
; SIDE EFFECTS:
; RESTRICTIONS:
; PROCEDURE:
;   Using bisection, find the root of the function x-t thought to lie
;   between the limits of the array. The root is refined until its
;   accuracy is +-1. (fashioned after FUNCTION RTSEC of the Numerical Recipes)
```

```

; MODIFICATION HISTORY:
;   Author: Richard Wagener, SAL/ISTS, April, 1989.
;   Documenation added: Ir lait 910225
; Bug fixed where array was double-valued function, A. Graps Nasa-Ames 11/93
;-
func=x-t
maxit=30 ; Maximum allowed number of iterations
x1=0
x2=n_elements(x)-1
f=func(x1)
fmid=func(x2)
minf=min(x)
maxf=max(x)
case 1 of
  t lt minf: begin
    print, 'Indx Error: Input val less than min(array)'
    return, x1
  end
  t gt maxf: begin
    print, 'Indx Error: Input val greater than max(array)'
    return, x2
  end
else:
endcase
if f lt 0 then begin
  rtbis=x1
  dx=x2-x1
endif else begin
  rtbis=x2
  dx=x1-x2
endelse
for j=1,maxit do begin
  dx=dx*0.5
  xmid=long(rtbis+dx+0.5)
  fmid=func(xmid)
  if fmid le 0 then rtbis=xmid
  if abs(dx) le 1 or fmid eq 0 then return,rtbis
endfor
print,'ERROR in INDX: Maximum number of iterations ',maxit,' reached.'
end

```

--

Amara Graps email: agraps@netcom.com
Computational Physics vita: finger agraps@sunshine.arc.nasa.gov
Multiplex Answers URL: http://www.amara.com/

"A committee is a life form with six or more legs and no brain."
--Lazarus Long
