Subject: search routine
Posted by Laurens on Fri, 01 Jun 2007 10:47:28 GMT
View Forum Message <> Reply to Message

Hi folks,

From Martin Schultz (posted in 1999) I found the following array-search algorithm which seems to do a fine job.
Except that i'm not able to catch the first element in the array.

Example:

Array = [0,80,100,120,180,300]
result = search, Array, 4.53

It should return index 0, if I understand it correctly, but it returns 1 instead. Now I don't quite follow the logic of the function, so maybe someone for which it's easy to see can help me in the right direction?

```
>  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

Thnx! Laurens

Subject: Re: search routine
Posted by cmancone on Fri, 01 Jun 2007 14:58:57 GMT
View Forum Message <> Reply to Message

On Jun 1, 10:27 am, Paolo Grigis <pgri...@astro.phys.ethz.ch> wrote:
> Laurens wrote:
>> Paolo Grigis wrote:
>
>>> Laurens wrote:
>>>> Hi folks,
>
>>>> From Martin Schultz (posted in 1999) I found the following
>>>> array-search algorithm which seems to do a fine job.
>>>> Except that i'm not able to catch the first element in the array.
>
>>>> Example:
>
>>>> Array = [0,80,100,120,180,300]
>>>> result = search, Array, 4.53
>
>>>> It should return index 0, if I understand it correctly, but it
>>>> returns 1 instead. Now I don't quite follow the logic of the
>>>> function, so maybe someone for which it's easy to see can help me in
>>>> the right direction?
>
>>> You could use the built-in function value_locate instead:
>
>>> result=value_locate(array,4.53)
>
>>> which returns 0.
>
>>> Ciao,
>>> Paolo
>
>> though i'm noticing that it takes its ground number to be returned.
>> If my array has [0,10,20,30] and i'm searching for 18, it will return
>> 10. Now its just that would want to get 20 instead of 10 :)
>
> well, in that case, just increase the index of the returned
> value by one (though you'd better check that it wasn't the
> *last* element...).

>
> Ciao,
> Paolo
>
>
>
>> regards, Laurens

Just to make sure it's clear, built in functions (value_locate) are always better - I was just pondering how it might be done if there wasn't a built in function. It's always fun to work out a problem anyway, even if there's already a correct answer.

Subject: Re: search routine
Posted by Laurens on Fri, 01 Jun 2007 15:24:37 GMT
View Forum Message <> Reply to Message

Paolo Grigis wrote:
>
>
> Laurens wrote:
>> Paolo Grigis wrote:
>>>
>>>
>>> Laurens wrote:
>>>> Hi folks,
>>>>
>>>> From Martin Schultz (posted in 1999) I found the following
>>>> array-search algorithm which seems to do a fine job.
>>>> Except that i'm not able to catch the first element in the array.
>>>>
>>>> Example:
>>>>
>>>> Array = [0,80,100,120,180,300]
>>>> result = search, Array, 4.53
>>>>
>>>> It should return index 0, if I understand it correctly, but it
>>>> returns 1 instead. Now I don't quite follow the logic of the
>>>> function, so maybe someone for which it's easy to see can help me in
>>>> the right direction?
>>>
>>> You could use the built-in function value_locate instead:
>>>
>>> result=value_locate(array,4.53)
>>>
>>> which returns 0.
>>>

>>> Ciao,
>>> Paolo
>>>
>> though i'm noticing that it takes its ground number to be returned.
>> If my array has [0,10,20,30] and i'm searching for 18, it will return
>> 10. Now its just that would want to get 20 instead of 10 :)
>
> well, in that case, just increase the index of the returned
> value by one (though you'd better check that it wasn't the
> *last* element...).
>
> Ciao,
> Paolo
>
>>
>> regards, Laurens
haha, that sounds like a plausible solution, but that would get a result
of 13 to become the 3rd index (2), when it should be the second.. It
just needs to mean the result before it compares the value, but i assume
that's not something which can be done without rewriting it?

Cheers, Laurens

Subject: Re: search routine
Posted by Paolo Grigis on Fri, 01 Jun 2007 15:39:53 GMT
View Forum Message <> Reply to Message

cmancone@ufl.edu wrote:
> On Jun 1, 10:27 am, Paolo Grigis <pgri...@astro.phys.ethz.ch> wrote:
>
> [...]
>
>
> Just to make sure it's clear, built in functions (value_locate) are
> always better

Oh, you're lucky to have such a faith! ;-)
I am a bit more skeptical about that...

I think it is a good thing to ponder the virtue of
routines that comes with IDL, and sometimes
they *do* leave some space for improvement...

> I was just pondering how it might be done if there
> wasn't a built in function. It's always fun to work out a problem
> anyway, even if there's already a correct answer.

Yes, I totally agree, the problem is that usually when
I do try and reinvent the wheel, it come out square-shaped.
But bumpy rides can be fun too :-)

Ciao,
Paolo
