In article <60a53808-3670-43e1-85da-ee00537487f9@googlegroups.com>,
 Craig Markwardt <craig.markwardt@gmail.com> wrote:

> On Friday, June 22, 2012 12:06:15 PM UTC-4, (unknown) wrote:
>> Now I find that it is not exactly what I'm looking for
>>
>> Suppose my grid is [5,1,12] and I want to find to which of these values a
>> data point of 4 is closest to.
>>
>> So I write
>>
>> grid = [5,1,12]
>> print, VALUE_LOCATE(grid,4)
>>   1
>>
>> But indeed it should be 0 since the 5 in the grid is closer to my data
>> point...
>> So in fact I need the nearest neighbor... :(
>
> By the way, your grid has to be strictly ascending. If you pass a randomly
> ordered grid, expect random results.
>
> VALUE_LOCATE() always finds the next lowest grid point, not the nearest
> gridpoint.
>
> On the other hand, it's easy enough to check for this.
>
>   x = your data points
>   grid = [1, 5, 12]
>   ii = value_locate(grid, x) ;; You already know this much
>
>   ;; See if the ii+1 grid point is closer
>   ;;   _no overflow_   ___ ii+1 sep ___   __ ii sep __
>   wh = where(ii LT 2   AND (grid[ii+1] - x) LT (x-grid[ii]), ct)
>
>   ;; If we found some, then use those instead
>   if ct GT 0 then ii[wh] = ii[wh]+1
>
> Craig

What he said!

Ken