Chong Frank Chowder
APCS2 pd02
HW02 – Speaking in Pseudocode
2017-02-01

Array Search Algorithm in O(n) Time

At first, our group believed it impossible to create an algorithm that would find a given search term in O(n) time. However, after melting down our brains into a potent chowder of knowledge, we have devised an algorithm that does precisely what we had thought impossible. Recipe below.

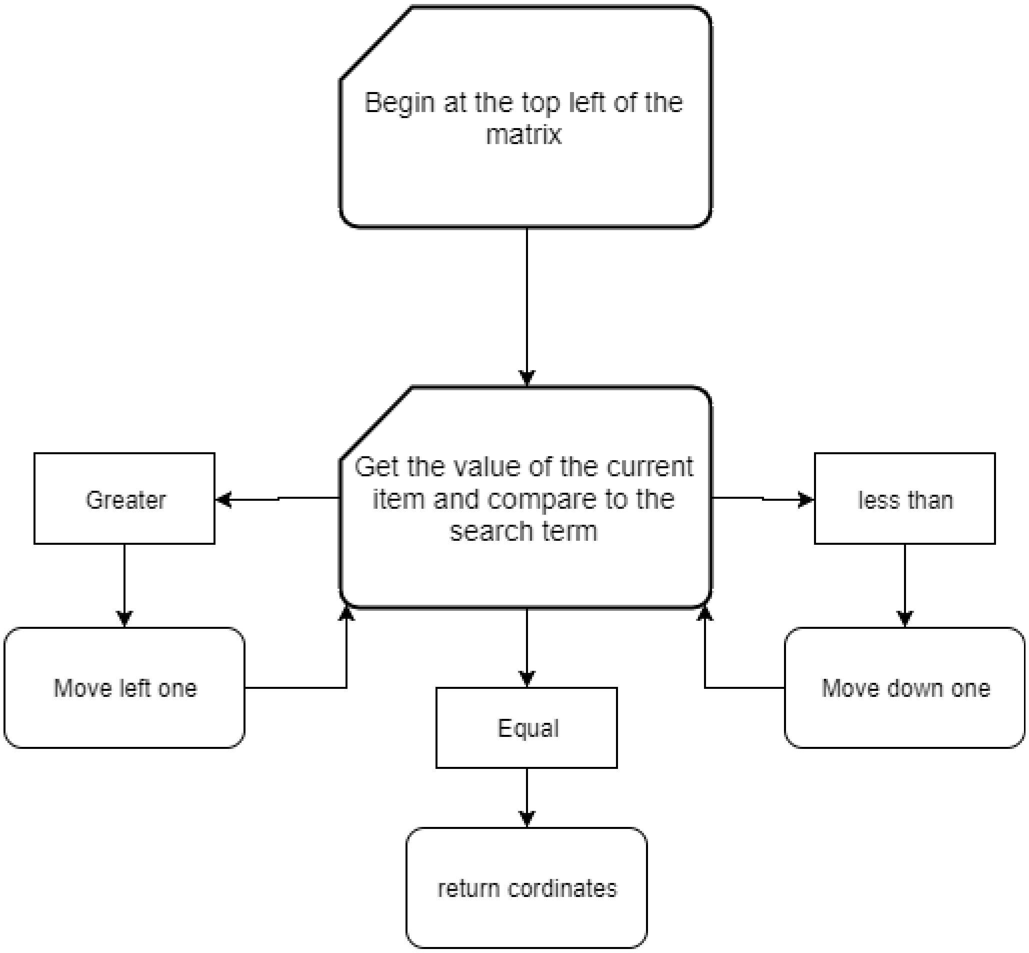
Step-by-Step
1. Using the matrix below, start at the top right item. Assume we are searching for 6.
1 2 3 **4**
2 3 4 5
3 4 5 7
4 5 6 8

2. Then we check if the current number is greater than, equal to, or smaller than the search number.
If equal: return coords, you've got it!
If greater: move left one
If less: move down one
In this case, it's less, so we'll move down
1 2 3 4
2 3 4 **5**
3 4 5 7
4 5 6 8
3. Repeat step 2 until finished
1 2 3 4   1 2 3 4   1 2 3 4
2 3 4 5 => 2 3 4 5 => 2 3 4 5
3 4 5 **7** => 3 4 **5** 7 => 3 4 5 7
4 5 6 8   4 5 6 8   4 5 **6** 8
4. Fin!

Why this is O(n)

The worst case for this array is if the item is in the bottom right corner. This is the farthest away the item could be from the start. If this is the case, the runtime is 2n, which is on the order of n.

Flow Chart:



Begin at the top left of the matrix
Greater
Get the value of the current item and compare to the search term
less than
Move left one
Move down one
Equal
return cordinates

And finally, some python code (it’s almost pseudocode):

```
#main function
def arraySearch(array, search):
    arraySearchHelper(array, search, len(array)-1, 0) #start the
search on the upper right corner

#Recursive helper for main function
def arraySearchHelper(array, search, x, y):
    currentItem = array[y][x] #store the current item
    if currentItem == search: #if it's found the value
        print(x,y) #print the coords
    elif currentItem > search: #if the current item is larger
then the search
        arraySearchHelper(array, search, x-1, y) #move to the
next item to the left
    else: #if the current item is smaller than the search
        arraySearchHelper(array, search, x, y+1) #move down one
```