1.4.17 Farthest pair (in one dimension). Write a program that, given an array a[] of N double values, finds a farthest pair : two values whose difference is no smaller than the the difference of any other pair (in absolute value). The running time of your program should be *linear* in the worst case.

Solution: One soln could be to sort the array if unsorted, then subtract end index(largest num) with start index(smallest num). This requires $nlogn$ complexity to sort the array first with fastest available sorting algo. This is linearithmic, not linear. We need linear soln.

A soln with $O(N)$ complexity would be to run through the array to find min and max, which will represent the pair with farthest distance.

```
//Pseudocode: Find fartest pair
function farthestpair(arr[])

    currMin = arr[0]
    currMax = arr[0]

    pair = []
    for i in range(len(arr)):
        if(arr[i] < currMin):
            currMin = arr[i]
        else if(arr[i] > currMax):
            currMax = arr[i]

    pair[0] = currMin
    pair[1] = currMax
    return pair
```