

Assignment 1

Date Assigned: 08/31/2018

Due: Midnight 09/07/2018 on iLearn

Please read [turn-in checklist](#) at the end of this document before you start doing exercises.

Section 1: Pen-and-paper Exercises

1. Consider the following problem:

Input: an array, A, of n sorted integers (positive, negative, or 0) that
 $A[0] \leq A[1] \leq A[2] \leq \dots \leq A[n-2] \leq A[n-1]$

Output: re-arrange elements in A such that:

Element at even position (i.e., A[0], A[2]) are less than or equal to both of its neighbors

Element at odd position (i.e., A[1], A[3]) are greater than or equal to both of its neighbors

$A[0] \leq A[1] \geq A[2] \leq A[3] \geq A[4] \leq A[5] \dots$

Design an algorithm that solves this problem.

(i) describe the idea behind your algorithm in English (5 points);

The outer loop contains an integer of a value that is always one less than that of the inner loop in the method 'rearrange'. These two int's serve as the two positions in an array that the method compares. Since the elements at even positions must always be less than or equal to its neighbors while those at odd positions must be greater than or equal to their neighbors, the inner 'if' statement compares the two positions in the array so that each element at an even position is assured greater than its neighbors in odd positions.

(ii) provide pseudocode (10 points);

Input A[]

n = size(A)

Initial = A[0]; initial represents the first element contained in the array A[]

For x in [0, size(A)-1]

 For y in [1, size(A) -1]

 If A[x] > A[y]

 Initial = A[y]

 End if

 End for

End for

2. Consider the following problem:

Input: an array A of n integers (positive, negative, or 0), elements sorted in ascending order.

Output: if there exists a majority element.

An element is a majority if it appears more than $n/2$ times. For example, if the input list is:

{0, 0, 0, 0, 0, 0, 1, 1, 2, 4, 7}

The output should be 0, as 0 appears 6 times ($>n/2 = 11/2$ times).

However, if the input list is:

{0, 0, 0, 1, 1, 2, 3, 10, 10}

The majority element does not exist.

Design an algorithm that solves this problem.

(i) describe the idea behind your algorithm in English (5 points);

The outer loop uses the variable *i* to represent an element in the array, and a *currentCount* variable to represent whatever majority might be found. The inner loop uses the variable *x* to represent elements in the array, which the inner if statements then compare to *i* as well as *n*, which is the length of the array. For each value of *i* in the outer loop, *x* represents every value in the array in the inner loop. If *currentCount* ever exceeds the value of $n / 2$, the variable 'maximum' is assigned the value of *currentCount* so that it can later return true.

(ii) provide pseudocode (10 points);
Input A[]

maximum = 0; maximum represents the majority if one does exist
n = size(A)

```
For i in [0, size(A)-1]
    currentCount = 0; holds the count so a majority can be found if it exists
    For x in [0, size(A)-1]
        If A[i] = A[x]
            Increment currentCount
        End if
        If currentCount > n/2
            maximum = currentCount
        End if
    End for
    If maximum > n / 2
        return true
    End if
Return false
End for
```

Section 2: Java Implementation

3. Implement problem 1 in Java.

Note:

Find a file called Problem1.java in assignment 1 folder.

Complete the method of rearrange().

Test your method in the main method provided following the comments.

Important: In all of the assignments of this course, when you are asked to implement an algorithm for a problem, your code will be evaluated based on:

5 points - Execution

Each file must run without error or warning on valid input described in the main method provided.

5 points - Within Code Documentation

Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?

20 points - Correctness

Is the algorithm implemented correctly? Does your method pass the test?

4. Implement problem 2 in Java.

Note:

Find a file called Problem2.java in assignment 1 folder.

Complete the method of majority().

Test your method in the main method provided following the comments.

TURN-IN CHECKLIST:

1. **Answers to Section 1 (.doc/.txt/.pdf), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName_LastName_assignment_1'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**