Satvik Anand

404-823-011

Discussion 1E

**Project 4 Report**

Test Data

Functions:

locateMinimum

* (empty array, non-zero positive integer) - The function should be able to implement itself even if the array has no characters in it.
* (any array, negative number) – The array should return -1 if the size is negative.
* (any array, 0) – The function should return 0, as there are no values in an array that is empty.
* (any array, non-negative number) – The function should return the index of the value at which the minimum value occurs.
* (an array with duplicates, non-negative number) – The function should return the index of the first minimum value, and should ignore the others.

findLastOccurence

* (empty array, non-zero positive integer, target) – The function should be able to return -1 as the target string will not be found.
* (any array, negative number, any target string) – The function should return -1 if the size is negative.
* (an array with duplicates of the target string, valid positive integer, target string) – The function should return only the index of the highest value at which the target string occurs.

flipAround

* (empty array, non-zero positive integer) – The function should be able to return -1.
* (any array, negative number) – The function should return -1 if the size is negative.
* (an array with an even number of elements) – The counter should be able to stop when the loop reaches the middle.
* (an array with an odd number of elements) – The counter should be able to stop when the loop reaches the middle.

hasNoDuplicates

* (empty array, non-zero positive integer) – The function should be able to return true as there are no duplicates if the array is empty.
* (any array, negative number) – The function should return false if the size is negative.
* (any array, 0) – The function should be able to return true as there are no duplicates if the array is empty.
* (array with only one element, positive integer) – The function should return true as there are no duplicates.
* (array with duplicates, positive integer) – The function should return false as there are duplicates.
* (array without duplicates but more than one element, positive integer) – The function should return true as there are no duplicates.

voidUnionWithNoDuplicates

* (any array (with 2 duplicates), 4, any array (with one duplicate from the previous array, also duplicated in the previous array), 2, resulting array, size of the resulting array)
  + To test this function, you need to verify that the resulting size is correct and that any of the necessary positions that are to be replaced are replaced correctly.
  + Eg: resultingSize in the above function should be 4, and if you test the array, it should output the element in that position.

shiftRight

* (any array, 0, 0, placeholder) – The function should return 0 even though the numbers are 0.
* (any array, any positive integer, a negative integer, placeholder) – The function should return -1.
* (any array, a negative integer, a positive integer, placeholder) – The function should return -1.
* (any array, a positive integer, a positive integer greater than the size, placeholder) – The function should replace all the elements with the placeholder and return the shift amount.
* (empty array, any positive integer, any positive integer, placeholder) – The function should return 0.
* (non-empty, any positive, any positive integer, placeholder) – The function should return the shifted amount.

isInIncreasingOrder

* (an empty array, positive integer) – The function should return true.
* (a non-empty array, 0) – The function should return true.
* (a normal array in ascending order, positive integer) – The function should return true.
* (a normal array not in ascending order, positive integer) – The function should return false.