David Lane

004638553

Project 4 Report

1. The first thing I had trouble with on this project was the findLastOccurrence function. I was going through the array backwards so that when I found the target string I could just return an integer to exit my for loop, but when I did this I forgot that I needed my ending condition for my for loop to use a greater than sign rather than a less than sign like the usual forward moving for loop. This small mistake caused my function to build but not run for a while, but eventually I figured it out. The other function I had some trouble with was the unionWithNoDuplicates function. When I was adding strings from the arrays passed into the function to a temporary string within the function, I was having some trouble with cases of the empty string. I assumed that this had something to do with the initialization of the temporary string to all empty strings (which conflicted with my duplicate checker) so I instead initialized my whole temporary array to a string item that I doubted the professor would give us and would therefore be able to be replaced and by pass duplicate exemption. Not a perfect solution but effective enough.
2. Test Cases:

(for all “arr,” assume this is an array of strings of size n)

For locateMinimum:

(arr, 0) to make sure the if n <= 0 return -1 check is working properly

(arr, 5) a basic array to ensure that the function is working properly

([an array with duplicate minimum], 5) to make sure the function returns the lowest index of a minimum

For findLastOccurrence:

(arr, 0, targetString) to make sure the if n <= 0 return -1 check is working properly

(arr, 5, targetString) a basic array to ensure that the function is working properly

([an array with no targetString], 5, targetString) to make sure the function returns -1

([an array with duplicate targetString], 5, targetString) to make sure the function returns the largest index of a match

For flipAround

(arr, -1) to make sure the if n < 0 return -1 check is working properly

(arr, 5) and (arr, 6) to make sure function works and returns proper number for arrays with both an odd and even number of items

For hasNoDuplicates

(arr, -1) to make sure the if n < 0 return -1 check is working properly

([an array with no duplicates], 5) to make sure function returns true

([an array with duplicates], 5) to make sure the function returns false

For unionWithNoDuplicates:

-Cases where n1 or n2 are less than or equal to 0 to make sure resultingSize goes to -1

-cases with empty strings in either array

-case where there is a duplicate in the first array

-case where there is a duplicate in the second array

-case where the same string is present in both arrays

-case where there are instances of multiple duplicates

For shiftRight:

-cases where amount is less than zero, greater than n, or n is less than 0 should all return -1

- (arr, 6, 2, "foo") case to check that the function works and returns the proper integer

For isInIncreasingOrder:

-cases where n = 0, 1, (should each return true) and -1 (should return false)

-case in increasing order

-case not in increasing order