Luis Lopez

Programming Assignment 1 Report

Part I.a

1. First, my strategy of finding common elements across the two arrays was to traverse once through both of my tally arrays simultaneously and comparing if both values equaled to 1, then I can get started on removing the common value and decreasing the element sizes of both ListA and ListB.
2. Second, the number of times I have to iterate through both arrays will depend on how many common values are present. For example, if two values are common on both arrays, then I will have to traverse through each array twice. That means I will have to iterate through each array (elementSize * 2 - 1) times. Minus 1 because the elementSize shrinks by one after the first time I traverse through the array.
3. Other "costs" that in my strategy that may incur is partially traversing through each array as I'm shifting the elements to the left to "delete" the common value found. For example, if one common element found in ListA is located at the ListA[0], then I will have to iterate through the rest of the elements and shift them to the left to cover up the "hole" made after "deletion."

Part I.b

1. When displaying the common elements of the two arrays in order, my strategy of using tally arrays played a very crucial role in making this happen. When traversing through the tally arrays simultaneously to find the common value, I simply printed out that element where both tallies equaled to 1. Since tally arrays are already ordered by the elements' nature, it became easier to display the common values in order.

Luis Lopez

Part I.c

2. To manually combine both arrays into one, I first created a new array size value as the sum of both the arrays sizes called combinedElementSize, and I also created a temporary position integer value to keep track of the element position of the new combined array while merging. Starting with the first ListA, I set up a for loop with its own size and set each element from ListA to the new TheListA array and incrementing the position value. After finishing populating TheListA array with values from ListA, I can now repeat the same process with ListB, and with the position value, I can continue to populate TheListA array where I left of. This strategy of combining both arrays is a bit costly because I will have to iterate through both arrays. But this cost is greatly reduced when its ready to be sorted. I used counting sort algorithm here to be able to sort the bigger sized array without comparing arrays values. The input array I used is TheListA, the output array is TheList, *k* is the maxVal from TheListA array and lastly, *n* is the array size combinedElementSize. One issue I could not figure out is the first zero that appears in the output array TheList.