CSC6013 - Worksheet for Week 7

Array Sum

1. Trace the recursive array sum algorithm for the following arrays. Show to each recursive call the input array, the returned value, and the number of sums executed. At the end of the trace, present the total number of sums executed (the total of sums of all recursive calls.

   a. **A = [38, 21, 39, 60, -1, 10, 81, 23]**

```
Split Array: Input Array = [38, 21, 39, 60, -1, 10, 81, 23] -> Output Arrays = [38, 21, 39, 60] and [-1, 10, 81, 23]
Split Array: Input Array = [38, 21, 39, 60] -> Output Arrays = [38, 21] and [39, 60]
Split Array: Input Array = [38, 21] -> Output Arrays = [38] and [21]
Base Case Reached: A[0,0] = 38 returned
Base Case Reached: A[1,1] = 21 returned
Sum: 38 + 21 = 59 returned | Total Sums = 1
Split Array: Input Array = [39, 60] -> Output Arrays = [39] and [60]
Base Case Reached: A[2,2] = 39 returned
Base Case Reached: A[3,3] = 60 returned
Sum: 39 + 60 = 99 returned | Total Sums = 2
Sum: 59 + 99 = 158 returned | Total Sums = 3
Split Array: Input Array = [-1, 10, 81, 23] -> Output Arrays = [-1, 10] and [81, 23]
Split Array: Input Array = [-1, 10] -> Output Arrays = [-1] and [10]
Base Case Reached: A[4,4] = -1 returned
Base Case Reached: A[5,5] = 10 returned
Sum: -1 + 10 = 9 returned | Total Sums = 4
Split Array: Input Array = [81, 23] -> Output Arrays = [81] and [23]
Base Case Reached: A[6,6] = 81 returned
Base Case Reached: A[7,7] = 23 returned
Sum: 81 + 23 = 104 returned | Total Sums = 5
Sum: 9 + 104 = 113 returned | Total Sums = 6
Sum: 158 + 113 = 271 returned | Total Sums = 7

Sum of Array: 271 | Total Sums Executed: 7
```

   b. **B = [2, 97, 5, 88, 9, 72, 12, 64, 17, 56, 21]**

```
Split Array: Input Array = [2, 97, 5, 88, 9, 72, 12, 64, 17, 56, 21] -> Output Arrays = [2, 97, 5, 88, 9, 72] and [12, 64, 17, 56, 21]
Split Array: Input Array = [2, 97, 5, 88, 9, 72] -> Output Arrays = [2, 97, 5] and [88, 9, 72]
Split Array: Input Array = [2, 97, 5] -> Output Arrays = [2, 97] and [5]
Split Array: Input Array = [2, 97] -> Output Arrays = [2] and [97]
Base Case Reached: A[0,0] = 2 returned
Base Case Reached: A[1,1] = 97 returned
Sum: 2 + 97 = 99 returned | Total Sums = 1
Base Case Reached: A[2,2] = 5 returned
Sum: 99 + 5 = 104 returned | Total Sums = 2
Split Array: Input Array = [88, 9, 72] -> Output Arrays = [88, 9] and [72]
Split Array: Input Array = [88, 9] -> Output Arrays = [88] and [9]
Base Case Reached: A[3,3] = 88 returned
Base Case Reached: A[4,4] = 9 returned
Sum: 88 + 9 = 97 returned | Total Sums = 3
Base Case Reached: A[5,5] = 72 returned
Sum: 97 + 72 = 169 returned | Total Sums = 4
Sum: 104 + 169 = 273 returned | Total Sums = 5
Split Array: Input Array = [12, 64, 17, 56, 21] -> Output Arrays = [12, 64, 17] and [56, 21]
Split Array: Input Array = [12, 64, 17] -> Output Arrays = [12, 64] and [17]
Split Array: Input Array = [12, 64] -> Output Arrays = [12] and [64]
Base Case Reached: A[6,6] = 12 returned
Base Case Reached: A[7,7] = 64 returned
Sum: 12 + 64 = 76 returned | Total Sums = 6
Base Case Reached: A[8,8] = 17 returned
Sum: 76 + 17 = 93 returned | Total Sums = 7
Split Array: Input Array = [56, 21] -> Output Arrays = [56] and [21]
Base Case Reached: A[9,9] = 56 returned
Base Case Reached: A[10,10] = 21 returned
Sum: 56 + 21 = 77 returned | Total Sums = 8
Sum: 93 + 77 = 170 returned | Total Sums = 9
Sum: 273 + 170 = 443 returned | Total Sums = 10

Sum of Array: 443 | Total Sums Executed: 10
```

c. **C = [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85]**

```
Split Array: Input Array = [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85] -> Output Arrays = [100, 33, 22, 213, 65, 29] and [153, 199, 47, 181, 85]
Split Array: Input Array = [100, 33, 22, 213, 65, 29] -> Output Arrays = [100, 33, 22] and [213, 65, 29]
Split Array: Input Array = [100, 33, 22] -> Output Arrays = [100, 33] and [22]
Split Array: Input Array = [100, 33] -> Output Arrays = [100] and [33]
Base Case Reached: A[0,0] = 100 returned
Base Case Reached: A[1,1] = 33 returned
Sum: 100 + 33 = 133 returned | Total Sums = 1
Base Case Reached: A[2,2] = 22 returned
Sum: 133 + 22 = 155 returned | Total Sums = 2
Split Array: Input Array = [213, 65, 29] -> Output Arrays = [213, 65] and [29]
Split Array: Input Array = [213, 65] -> Output Arrays = [213] and [65]
Base Case Reached: A[3,3] = 213 returned
Base Case Reached: A[4,4] = 65 returned
Sum: 213 + 65 = 278 returned | Total Sums = 3
Base Case Reached: A[5,5] = 29 returned
Sum: 278 + 29 = 307 returned | Total Sums = 4
Sum: 155 + 307 = 462 returned | Total Sums = 5
Split Array: Input Array = [153, 199, 47, 181, 85] -> Output Arrays = [153, 199, 47] and [181, 85]
Split Array: Input Array = [153, 199, 47] -> Output Arrays = [153, 199] and [47]
Split Array: Input Array = [153, 199] -> Output Arrays = [153] and [199]
Base Case Reached: A[6,6] = 153 returned
Base Case Reached: A[7,7] = 199 returned
Sum: 153 + 199 = 352 returned | Total Sums = 6
Base Case Reached: A[8,8] = 47 returned
Sum: 352 + 47 = 399 returned | Total Sums = 7
Split Array: Input Array = [181, 85] -> Output Arrays = [181] and [85]
Base Case Reached: A[9,9] = 181 returned
Base Case Reached: A[10,10] = 85 returned
Sum: 181 + 85 = 266 returned | Total Sums = 8
Sum: 399 + 266 = 665 returned | Total Sums = 9
Sum: 462 + 665 = 1127 returned | Total Sums = 10

Sum of Array: 1127 | Total Sums Executed: 10
```

Mergesort

2. Trace the Mergesort algorithm for the following arrays. Show to each recursive call the two input and output arrays.
   a. **A = [38, 21, 39, 60, -1, 10, 81, 23]**

```
Split Array: Input Array = [38, 21, 39, 60, -1, 10, 81, 23] -> Output Arrays = [38, 21, 39, 60] and [-1, 10, 81, 23]
Split Array: Input Array = [38, 21, 39, 60] -> Output Arrays = [38, 21] and [39, 60]
Split Array: Input Array = [38, 21] -> Output Arrays = [38] and [21]
Base Case Reached: 38 returned
Base Case Reached: 21 returned
Merge Arrays: Input Arrays = [38] and [21] -> Output Array = [21, 38]
Split Array: Input Array = [39, 60] -> Output Arrays = [39] and [60]
Base Case Reached: 39 returned
Base Case Reached: 60 returned
Merge Arrays: Input Arrays = [39] and [60] -> Output Array = [39, 60]
Merge Arrays: Input Arrays = [21, 38] and [39, 60] -> Output Array = [21, 38, 39, 60]
Split Array: Input Array = [-1, 10, 81, 23] -> Output Arrays = [-1, 10] and [81, 23]
Split Array: Input Array = [-1, 10] -> Output Arrays = [-1] and [10]
Base Case Reached: -1 returned
Base Case Reached: 10 returned
Merge Arrays: Input Arrays = [-1] and [10] -> Output Array = [-1, 10]
Split Array: Input Array = [81, 23] -> Output Arrays = [81] and [23]
Base Case Reached: 81 returned
Base Case Reached: 23 returned
Merge Arrays: Input Arrays = [81] and [23] -> Output Array = [23, 81]
Merge Arrays: Input Arrays = [-1, 10] and [23, 81] -> Output Array = [-1, 10, 23, 81]
Merge Arrays: Input Arrays = [21, 38, 39, 60] and [-1, 10, 23, 81] -> Output Array = [-1, 10, 21, 23, 38, 39, 60, 81]

Final Sorted Array: [-1, 10, 21, 23, 38, 39, 60, 81]
```

b. B = [2, 97, 5, 88, 9, 72, 12, 64, 17, 56, 21]

```
Split Array: Input Array = [2, 97, 5, 88, 9, 72, 12, 64, 17, 56, 21] -> Output Arrays = [2, 97, 5, 88, 9] and [72, 12, 64, 17, 56, 21]
Split Array: Input Array = [2, 97, 5, 88, 9] -> Output Arrays = [2, 97] and [5, 88, 9]
Split Array: Input Array = [2, 97] -> Output Arrays = [2] and [97]
Base Case Reached: 2 returned
Base Case Reached: 97 returned
Merge Arrays: Input Arrays = [2] and [97] -> Output Array = [2, 97]
Split Array: Input Array = [5, 88, 9] -> Output Arrays = [5] and [88, 9]
Base Case Reached: 5 returned
Split Array: Input Array = [88, 9] -> Output Arrays = [88] and [9]
Base Case Reached: 88 returned
Base Case Reached: 9 returned
Merge Arrays: Input Arrays = [88] and [9] -> Output Array = [9, 88]
Merge Arrays: Input Arrays = [5] and [9, 88] -> Output Array = [5, 9, 88]
Merge Arrays: Input Arrays = [2, 97] and [5, 9, 88] -> Output Array = [2, 5, 9, 88, 97]
Split Array: Input Array = [72, 12, 64, 17, 56, 21] -> Output Arrays = [72, 12, 64] and [17, 56, 21]
Split Array: Input Array = [72, 12, 64] -> Output Arrays = [72] and [12, 64]
Base Case Reached: 72 returned
Split Array: Input Array = [12, 64] -> Output Arrays = [12] and [64]
Base Case Reached: 12 returned
Base Case Reached: 64 returned
Merge Arrays: Input Arrays = [12] and [64] -> Output Array = [12, 64]
Merge Arrays: Input Arrays = [72] and [12, 64] -> Output Array = [12, 64, 72]
Split Array: Input Array = [17, 56, 21] -> Output Arrays = [17] and [56, 21]
Base Case Reached: 17 returned
Split Array: Input Array = [56, 21] -> Output Arrays = [56] and [21]
Base Case Reached: 56 returned
Base Case Reached: 21 returned
Merge Arrays: Input Arrays = [56] and [21] -> Output Array = [21, 56]
Merge Arrays: Input Arrays = [17] and [21, 56] -> Output Array = [17, 21, 56]
Merge Arrays: Input Arrays = [12, 64, 72] and [17, 21, 56] -> Output Array = [12, 17, 21, 56, 64, 72]
Merge Arrays: Input Arrays = [2, 5, 9, 88, 97] and [12, 17, 21, 56, 64, 72] -> Output Array = [2, 5, 9, 12, 17, 21, 56, 64, 72, 88, 97]

Final Sorted Array: [2, 5, 9, 12, 17, 21, 56, 64, 72, 88, 97]
```

c. C = [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85]

```
Split Array: Input Array = [100, 33, 22, 213, 65, 29, 153, 199, 47, 181, 85] -> Output Arrays = [100, 33, 22, 213, 65] and [29, 153, 199, 47, 181, 85]
Split Array: Input Array = [100, 33, 22, 213, 65] -> Output Arrays = [100, 33] and [22, 213, 65]
Split Array: Input Array = [100, 33] -> Output Arrays = [100] and [33]
Base Case Reached: 100 returned
Base Case Reached: 33 returned
Merge Arrays: Input Arrays = [100] and [33] -> Output Array = [33, 100]
Split Array: Input Array = [22, 213, 65] -> Output Arrays = [22] and [213, 65]
Base Case Reached: 22 returned
Split Array: Input Array = [213, 65] -> Output Arrays = [213] and [65]
Base Case Reached: 213 returned
Base Case Reached: 65 returned
Merge Arrays: Input Arrays = [213] and [65] -> Output Array = [65, 213]
Merge Arrays: Input Arrays = [22] and [65, 213] -> Output Array = [22, 65, 213]
Merge Arrays: Input Arrays = [33, 100] and [22, 65, 213] -> Output Array = [22, 33, 65, 100, 213]
Split Array: Input Array = [29, 153, 199, 47, 181, 85] -> Output Arrays = [29, 153, 199] and [47, 181, 85]
Split Array: Input Array = [29, 153, 199] -> Output Arrays = [29] and [153, 199]
Base Case Reached: 29 returned
Split Array: Input Array = [153, 199] -> Output Arrays = [153] and [199]
Base Case Reached: 153 returned
Base Case Reached: 199 returned
Merge Arrays: Input Arrays = [153] and [199] -> Output Array = [153, 199]
Merge Arrays: Input Arrays = [29] and [153, 199] -> Output Array = [29, 153, 199]
Split Array: Input Array = [47, 181, 85] -> Output Arrays = [47] and [181, 85]
Base Case Reached: 47 returned
Split Array: Input Array = [181, 85] -> Output Arrays = [181] and [85]
Base Case Reached: 181 returned
Base Case Reached: 85 returned
Merge Arrays: Input Arrays = [181] and [85] -> Output Array = [85, 181]
Merge Arrays: Input Arrays = [47] and [85, 181] -> Output Array = [47, 85, 181]
Merge Arrays: Input Arrays = [29, 153, 199] and [47, 85, 181] -> Output Array = [29, 47, 85, 153, 181, 199]
Merge Arrays: Input Arrays = [22, 33, 65, 100, 213] and [29, 47, 85, 153, 181, 199] -> Output Array = [22, 29, 33, 47, 65, 85, 100, 153, 181, 199, 213]

Final Sorted Array: [22, 29, 33, 47, 65, 85, 100, 153, 181, 199, 213]
```