CSCE4110.002/005

3-7-21

Group: Andrew Araujo, Andrew Ragland, Sanjul Sharma, Samuel Holsomback

Assignment #3

1. (10 pts.) Use iterative method (recursion tree method) to compute the following recursive functions

a.T(n) = T(n-1) + T(n-2) + n

b.T(n) = 2T(n-1) + n2

2. (10 pts.) Use Master Theorem method to solve the following recursive functions

a.T(n) = 4T(n/2) + n2logn

b.T(n) = 2T(n/4) + n2

c.T(n) = 2T(n/4) + 𝑛1/2

3. (10 pts.) Use Strassen’s algorithm to compute the product of the following matrices. Show the details as we did in class.

()

4. Implement a program to compute the minimum and maximum elements in an array using divide and conquer approach. Your program should divide the array into 2 subarrays and recursively call the operation on both arrays. After the recursive calls are computed, it should “combine” the results and report the final result.

Example:

Input: [5, 7, 2, 4, 9, 6]

Output:

The minimum array element is 2

The maximum array element is 9

1. (10 pts.) Write the pseudocode of your algorithm
2. .b. (5 pts.) Write the recurrence relation of your pseudocode as T(n) and solve it using one of the methods we discussed in class (i.e., substitution method, iteration method or master theorem).
3. c. (20 pts.) Implement your algorithm in C++. Make sure your program is well documented with indentation, comments and variable names. Your programshould read input from your code only. Include the input array given below in the main function in your code and then work on it to find the solution but make sure the code works for various inputs.

Input array = [5, 7, 2, 4, 3, 8, 9, 6]

5. Inversion Count for an array indicates how far/close the array is from being sorted. If array is already sorted then inversion count is 0. If array is sorted in reverse order that inversion count is the maximum. In this problem, given an array you will design an algorithm to compute the inversion count of that array.

Examples:

Input array: [8, 4, 2, 1]

Output: 6

Explanation: The array has six inversions:

(8,4), (4,2), (8,2), (8,1), (4,1), (2,1)

Input array: [3, 1, 2]

Output: 2

Explanation: The array has two inversions:

(3,1), (3,2)

1. (25 pts.) Write a pseudocode for a divide and conquer algorithm to compute inversion count of an input array.

inverseCount(array[])

{  
 if array[].length > 1:

inverseCount()

Else

}

b. (10 pts.) Write the recurrence relation of your pseudocode as T(n) and solve it using one of the methods we discussed in class (i.e., substitution method, iteration method or master theorem).

For full credit, your algorithm must be time algorithm.Θ(n𝑙𝑜𝑔𝑛)

**Hint**: Approach is similar to merge sort.