CIS 606 Spring 2022
Homework 2
Due on Feb 14th, 2022 (in class)

1. Design a linear-time, non-recursive algorithm for the Maximum-subarray problem discussed in our lecture. Given an array A[1,…,n], start at the left end of the array, and progress toward the right, keeping track of the maximum subarray seen so far. Suppose a maximum-subarray of A[1,…,j] is known, the maximum-subarray ending at index j+1 should be either a maximum-subarray of A[1,…,j] or a subarray A[i,…,j+1], for some $1 \le i \le j+1$. A key observation: determine a maximum-subarray of A[i,…,j+1] costs a constant time if a maximum subarray of A[i,…,j] is known. Hint: you may want to construct a table to store the key information of a maximum-subarray of A[1,…,j], as j progresses from 1 to n. Please write down the pseudo-code that describes the algorithm.

2. Prove by Mathematical Induction:
a) Given $T(n) = T(n-1) + n, T(n) \in O(n^2)$.
b) Given $T(n) = T(\lceil n/2 \rceil) + 1, T(n) \in O(logn)$.
c) Given $T(n) = 2T(\lfloor n/2 \rfloor) + n, T(n) \in \Omega(nlog\, n)$.

3. Using the master method to show that the solution to the recurrence T(n)=4T(n/3)+n is T(n)∈ $\theta(n^{\log_3 4})$.

4. Given the solution to problem 3, try to prove the same by using the substitution method. In particular, show that a substitution proof with the assumption T(n)≤ $cn^{\log_3 4}$ fails. Then show how to subtract off a low-order term to make a substitution proof work.

5. Use a recursion tree to determine a good asymptotic upper bound on the recurrence T(n)=T(n-1)+T(n/2)+n. Then, use the substitution method to verify your answer.