Introduction to Algorithms

Homework 1

Due 9/27/2018 in class

This part of homework is not for programming but helps clarify some ideas and concepts. Hand in your homework on due day **before** the class starts. Late homework will not be accepted as usual.

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

2. Use the master method to give tight asymptotic bounds for the following recurrences. If the master method does not work, explain why.
   (a) T(n) = 2T(n/4) + $\sqrt{n}$.
   (b) T(n) = 4T(n/2) + $\mathrm{n}^2 \log n$

3. Use the following ideas to develop a non-recursive, linear-time algorithm for the maximum-subarray problem. Start at the left end of the array, and progress toward the right, keeping track of the maximum subarray seen so far. Knowing a maximum subarray of A[1..j], extend the answer to find a maximum subarray ending at index j+1 by using the following observation: a maximum subarray of A[1..j +1] is either a maximum subarray of A[1..j] or a subarray A[i..j + 1], for some$1 \leq \mathrm{i} \leq \mathrm{j} + 1$. Determine a maximum subarray of the form A[i..j + 1] in constant time based on knowing a maximum subarray ending at index j.

4. Problem 4-5 on page 109 of the textbook.