Todd Wenker

CSE 310

Professor Nakamura

2/6/15

Homework Assignment 3

1. Suppose that T(n) = T(n/8) + T(7n/8) 5 n. Prove that T(n) = Θ(nlog n) by drawing a recursion tree. You need to show that T(n) = O(nlog n) and T(n) = Ω(nlog n) in order to show T(n) = Θ(nlog n). (logs are base 8).

T(n)

=

5n 5n

+ +

T(n/8) T(7n/8)

= =

5(n/8) 5(7n/8) 5n

+ + + +

T(n/64) T(7n/64) T(7n/64) T(49n/64)

5n

levels in the recursion tree (log base 8): (log n)

work done on each level: 5n

recursion equation (log base 8): (5n)(log n) = Θ(nlog n)

1. Suppose that the recurrence T(n) = T(n-2) + 2n +1. Use a use a recusion tree to give a asymptotically tight solution of T(n). You need to show Θ bound by computing its exact running time. Also assume that T(1) = 3.

T(n)

=

2n + 1

+

T(n-2)

=

2n+1

T(n) = T(1) + ∑(2\*k) +1

1. Let T(1) = 3 for n = 1, and for all n >= 2, T(n) = T(n-2) + 2n + 1. Solve this recurrence using the mathematical induction method. (i.e., You need to guess a solution – explicit solution of T(n) by specifying all coefficients, do not use O, Ω or Θ). Then assume that the guess is true for n = k, and prove that it is also true for n = k + 2. You also need to prove that your guess is true for the base case, n = 3 using the recurrence and your guess.
2. Use the master method to give tight asymptotic bouds for the following recurrences. (specify, a, b, f(n), and ε value when they apply.)
   1. T(n) = 4T(n/5) + n^2

a = 4, b = 5, f(n) = n^2

logb a = log5 4 < log5 5 = 1 f(n) = Ω(n^(logba + ε)) where ε = 1

Case 3: Regularity Condition: af(n/b) <= cf(n) for some constant c < 1

4\*(n/5)^2 <= cn^2 🡺 4/25 \*n^2 <= cn^2 if c = 4/25,

then 4/25n^2 <= 4/25n^2 for all n >= 1

so T(n) = Θ(f(n)) = Θ(n^2)

* 1. T(n) = 9T(n/3) + n^2

a = 9, b = 3, f(n) = n^2

log3 9 = 2, f(n) = Θ(n^ log3 9) = Θ(n^2)

Case 2: T(n) = Θ(n^ (log3 9) \* log n).

* 1. T(n) = 8T(n/2) +n

a = 8, b = 2, f(n) = n

log2 8 = 3, f(n) = O(n^(3 – ε)) where ε = 1 or f(n) = O(n^2)

Case 1: T(n) = Θ(n^(logb a)) = Θ(n^3).

