Matas Empakeris 11112142

Marika Hokhold

Assignment #4

1. T(n) = { 3T(4n/9) + 3T(n/9) + n1.5 if n ≥ 9

{ 1 if n ≤ 8

We will find the tight upper bound using a recurrence tree

\_\_\_\_\_\_\_ n1.5\_\_\_\_\_ ≤ cn

/ \

≤c(12n/9) (12n/9)1.5 (3n/9)1.5 ≤ c(3n/9)

/ \ / \

≤c(12n/9/9) (12n/81)1.5(12n/81)1.5 (3n/81)1.5 (3n/81)1.5 ≤ c(3n/9/9)

•

•

•

The longest path keeps reducing by a factor of 12/9ths so we can make a guess that this side is of length log9/12n. Thus we can say that the upper bound is O (n1.5 log(n)).

1. Both of these cases can be used with the Master Theorem. Using the definition of Master Theorem, we get that a is constant, a polynomial difference occurs between ρ(n)\*n2 and nlog\_b(a), a > 1, and f(n) will always result in a number > 1 if n >= 1.
2. T(n) = 2T(n/2) + n2ρ(n)

Applying the Master Theorem, we have a = 2, b = 2 and c = 2, so log2­2 = 0 which is less than 2. This satisfies case 1 and thus, T(n) = Θ(n2).

1. T(n) = 2T(n/3) + n2ρ(n)

Applying the Master Theorem, we have a = 2, b = 3, and c = 2, so log32 (which is approximately .6309), which is again less than 2. This satisfies case 1 and thus, T(n) = Θ(n2).