EECS340 - Algorithms - HW#3

Mark Schultz - mxs802

September 22, 2010

4.3-1

$T(n) = T(n-1) + n$ is $O(n^2)$

$$\begin{aligned} T(n-1) &\leq c(n-1)^2 \\ T(n) &\leq c(n-1) + n \\ T(n) &\leq c(n^2 - 2n + 1) + n = cn^2 - 2cn + c + n \\ T(n) &\leq cn^2 \end{aligned}$$

4.1

c) $T(n) = 16T(n/4) + n^2$
Here $a = 16, b = 4,$ and $f(n) = n^2$. Using the second case of the master method, $n^{\log_b a} = n^{\log_4 16}$ which simplifies to n^2. This fits the definition of the second case therefore $T(n) = \Theta(n^2 \lg n)$.

d) $T(n) = 7T(n/3) + n^2$
Here $a = 7, b = 3,$ and $f(n) = n^2$. To prove the third case we need $n^{\log_b a} = n^{\log_3 7}$ and because $1 < \log_3 7 < 2$ we know that $n^2 = \Omega(n^{\log_3 7 + \epsilon}$ for $\epsilon > 0$ and $7f(n/3) = 7(n/3)^2 = cf(n) = cn^2$ for $c = 7/9$ which is ¡ 1. Therefore $T(n) = \Theta(n^2)$.

e) $T(n) = 7T(n/2) + n^2$ // Here $a = 7, b = 2, f(n) = n^2$ and $n^{\log_b a} = n^{\lg 7}$. We can prove that $n^2 = O(n^{\log_2 7 - \epsilon})$ for $\epsilon > 0$ because $2 < \lg 7 < 3$. Therefore this falls under case 1 of the master method and $T(n) = \Theta(n^{\lg 7})$.

f) $T(n) = 2T(n/4) + \sqrt{n}$
Here $a = 2, b = 4,$ and $f(n) = \sqrt{n}$. because $\Theta(n^{\log_4 2}) = \sqrt{n}$ this falls under case 2 of the master theorem, therefore, $T(n) = \Theta(\sqrt{n} \lg n)$.