COSC 302: Analysis of Algorithms — Spring 2018
Adam Pettway
Colgate University

Problem Set 3 — Solving Recurrences and Divide and Conquer I

1. Problem 4-1 from CLRS 3rd edition.

   a The third case of the master method applies since $f(n)$ dominates the running time therefore it's $\theta(n^4)$.

   b The third case of the master method applies since $f(n)$ domintates the running time, therefore it's $\theta(n)$.

   c The second case of the master method applies since the running time is evenly distrubuted, therefore it is $\theta(n^2 log(n))$.

   d The third case of the master method applies since $f(n)$ dominates the running time, therefore it's $\theta(n^2)$.

   e The first case of the master method applies since $n^l og_2(7)$ dominates, therefore it's $\theta(n^{2.807})$.

   f The first case of the master method applies since the running time is evenly distrubuted, therefore it is $\theta(\sqrt{n} log(n))$.

   g The master method does not apply since there is no b value. Therefore we shall solve the recurrence by subsitution. First we start with the recurrence $T(n) = T(n-2) + n^2$. And we then subsitute $n-1$ into the recurrence to get $T(n-1) = T(n-3) + T(n-2)^2 + n^2$. It's clear that there will be n number of recursive calls therefore, the running time is $\theta(n^2)$.

2. Using one of the methods discussed in lecture, give a tight asymptotic bound for the recurrence $T(n) = 8T(n/2) + n^3 \log n$.
   The third case of the master method applies since $f(n)$ dominates the running time since $n^3 = O(n^3 \log(n))$. $f(n)$ also passes the regularity tests since $8\frac{n}{2}^3 log(\frac{n}{2}) \leq cn^3 \log(n)$ when $c = 8$.

3. Problem 4.3-9 from CLRS 3rd edition.

4. *Revisiting the pinePhone.* You are still hard at work testing the quality of pinePhones for Pineapple.

   (a) With 3 pinePhones, one can divide the ladder by $\sqrt[3]{n}$ parts. Then start at the highest rung of each part ($c * \sqrt[3]{n}$ with $1 \leq c \leq n/\sqrt[3]{n}$). If one of these highest rungs break then check the lower ladders by 1. Which would lead to a maximum of 3 phones broken.

   (b) YOne can find the highest safe rung with 4 pinePhones with $\Theta(\sqrt[4]{n})$ pinePhone drops. With 3 pinePhones, one can divide the ladder by $\sqrt[4]{n}$ parts. Then start at the highest rung of each part ($c * \sqrt[4]{n}$ with $1 \leq c \leq n/\sqrt[4]{n}$). If one of these highest rungs break then check the lower ladders by 1. Which would lead to a maximum of 4 phones broken.

(c) $\begin{cases} n & k = 1 \\ T(\sqrt[k]{n}) + \theta(1) & k \geq 2 \end{cases}$