COSC 302: Analysis of Algorithms — Spring 2018
Prof. Darren Strash
Colgate University

Problem Set 3 — Solving Recurrences and Divide and Conquer I

Due by 4:30pm Friday, Feb. 16, 2018 as a single pdf via Moodle (either generated via LaTeX, or concatenated photos of your work). Late assignments are not accepted.

This is an *individual* assignment: collaboration (such as discussing problems and brainstorming ideas for solving them) on this assignment is highly encouraged, but the work you submit must be your own. Give information only as a tutor would: ask questions so that your classmate is able to figure out the answer for themselves. It is unacceptable to share any artifacts, such as code and/or write-ups for this assignment. If you work with someone in close collaboration, you must mention your collaborator on your assignment.

Suggested practice problems (not to be turned in): 4.3-1, 4.3-8, 4.4-2, 4.4-4, 4.4-6, 4-3

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

   Give tight upper and lower bounds for the following recurrences.

   Solutions:

   (a) $T(n) = 2T(n/2) + n^4$.

   By case 3 of the Master Theorem $T(n) = \Theta(n^4)$.
   $n^{\log_b a} = n^{\log_2 2} = n$ and $f(n) = n^4 = \Omega(n^{1+\epsilon})$ for $0 < \epsilon \leq 3$. Furthermore, $f(n)$ satisfies the regularity condition with $c = 1/8$, since $2f(n/2) = 2\left(\frac{n}{2}\right)^4 = (1/8)n^4 \leq (1/8)f(n)$.

   (b) $T(n) = T(7n/10) + n$.

   By case 3 of the Master Theorem $T(n) = \Theta(n)$.
   $n^{\log_b a} = n^{\log_{10/7} 1} = n^0 = 1$ and $f(n) = n = \Omega(n^{0+\epsilon})$ for $0 < \epsilon \leq 1$. Furthermore, $f(n)$ satisfies the regularity condition with $c = 7/10$, since $1f(n/10/7) = n \leq 7/10n \leq 7/10f(n)$.

   (c) $T(n) = 16T(n/4) + n^2$.

   By case 2 of the Master Theorem $T(n) = \Theta(n^2 \lg n)$.
   $n^{\log_b a} = n^{\log_4 16} = n^2$ and $f(n) = n^2 = \Theta(n^2)$.

   (d) $T(n) = 7T(n/3) + n^2$.

   By case 3 of the Master Theorem $T(n) = \Theta(n^2)$.
   $n^{\log_b a} = n^{\log_3 7}$ and $f(n) = n^2 = \Omega(n^{\log_3 7+\epsilon})$ for $0 < \epsilon \leq 2 - \lg_3 7$.

See next page for finished solution →

(e) $T(n) = 7T(n/2) + n^2$.

By case 1 of the Master Theorem $T(n) = \Theta(n^{\log_2 7})$.
$n^{\log_b a} = n^{\log_2 7}$ and $f(n) = n^2 = O(n^{\log_2 7 - \epsilon})$ for $0 < \epsilon \leq \lg_2 7 - 2$.

(f) $T(n) = 2T(n/4) = \sqrt{n}$.

By case 2 of the Master Theorem $T(n) = \Theta(\sqrt{n} \lg n)$.
$n^{\log_b a} = n^{\log_4 2} = \sqrt{n}$ and $f(n) = \sqrt{n} = \Theta(\sqrt{n})$.

(g) $T(n) = T(n-2) + n^2$.

We show that $T(n) = \Theta(n^3)$ via the direct method.

Proof.

$$
\begin{aligned}
T(n) &= T(n-2) + n^2 \\
&= T(n-4) + (n-2)^2 + n^2 \\
&= T(n-6) + (n-4)^2 + (n-2)^2 + n^2 \\
&= T(n - 2(i+1)) + (n-2i)^2 + \cdots + n^2 \\
&= \Theta(1) + \sum_{i=0}^{n/2-1} (n-2i)^2 \\
&= \Theta(1) + \sum_{i=0}^{n/2-1} (n^2 - 4ni + 4i^2) \\
&= \Theta(1) + \sum_{i=0}^{n/2-1} n^2 - \sum_{i=0}^{n/2-1} 4ni + 4\sum_{i=0}^{n/2-1} 4i^2 \\
&= \Theta(1) + n^3/2 - 4n\frac{(n/2-1)(n/2)}{2} + 4\frac{(n/2)(n/2-1)(n-1)}{6} \\
&= \Theta(1) + n^3/2 - n^3/2 - \Theta(n^2) + n^3/6 + \Theta(n^2) \\
&= \Theta(n^3).
\end{aligned}
$$

□

2. Using one of the methods discussed in lecture, give a tight asymptotic bound for the recurrence $T(n) = 8T(n/2) + n^3 \lg n$.

Solution: As a first try, we can attempt to apply the master method. Unfortunately, it does not apply. Case 3 is the closest, but $n^3 \lg n \neq \Omega(n^{3+\epsilon})$, since $\lg n \leq n^\epsilon$ for any constant $\epsilon > 0$. We prove that $T(n) = \Theta(n^3 \lg^2 n)$ using the direct method.

Proof. By the proof of the master theorem, we have the

$$T(n) = \Theta(n^{\log_b a}) + \sum_{i=0}^{\log_b n - 1} a^i f(n/b^i),$$

which applies regardless of $f(n)$. Substituting in for $a, b, f(n)$, we have that

See next page for finished proof →

$$
\begin{aligned}
T(n) &= \Theta(n^{\log_b a}) + \sum_{i=0}^{\log_b n - 1} a^i f(n/b^i) \\
&= \Theta(n^{\log_2 8}) + \sum_{i=0}^{\lg n - 1} 8^i f(n/2^i) \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} 8^i \left(\frac{n}{2^i}\right)^3 \lg(n/2^i) \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} 8^i \left(\frac{n^3}{8^i}\right) \lg(n/2^i) \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} n^3 \lg(n/2^i) \qquad\qquad \text{(stop here for } T(n) = O(n^3 \lg^2 n)\text{)} \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} n^3 (\lg n - \lg 2^i) \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} n^3 (\lg n - i \lg 2) \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} (n^3 \lg n - n^3 i) \\
&= \Theta(n^3) + \sum_{i=0}^{\lg n - 1} n^3 \lg n - \sum_{i=0}^{\lg n - 1} n^3 i \\
&= \Theta(n^3) + n^3 \lg n \sum_{i=0}^{\lg n - 1} 1 - n^3 \sum_{i=0}^{\lg n - 1} i \\
&= \Theta(n^3) + n^3 \lg n \lg n - n^3 \frac{(\lg n - 1)\lg n}{2} \\
&= \Theta(n^3) + n^3 \lg n \left(\lg n - \frac{1}{2}(\lg n - 1)\right) \\
&= \Theta(n^3) + n^3 \lg n \left(\frac{1}{2}\lg n + \frac{1}{2}\right) \\
&= \Theta(n^3 \lg^2 n).
\end{aligned}
$$

$\square$

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

   Solve $T(n) = 3T(\sqrt{n}) + \log n$ by change of variables. Do not worry about whether values are integral.

   Solution:

   We let $n = 10^m$, and note that $m = \log n$. *(Note that we can also use $n = 2^m$ and get the same solution.)* Then we have that

$$T(10^m) = 3T(\sqrt{10^m}) + \log 10^m = 3T(10^{m/2}) + m.$$

   Notice that $T(10^m)$ is just a function of m. Let's rename it $S(m)$. Then,

$$S(m) = 3S(m/2) + m,$$

   which can be solved using case 1 of the master theorem. In this case $a = 3$, $b = 2$, and $f(m) = m$, and $m = O(m^{\log_2 3 - \epsilon})$ for any $0 < \epsilon \leq \log_2 3 - 1$.

   Therefore, $S(m) = \Theta(m^{\log_2 3})$, and since $m = \log n$, we have that

$$T(n) = T(10^m) = S(m) = \Theta(m^{\log_2 3}) = \Theta((\log n)^{\log_2 3}).$$

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

   (a) Suppose now that you are given 3 pinePhones. Present a strategy to find the highest safe rung with $\Theta(\sqrt[3]{n})$ pinePhone drops.

   Solution: Section the ladder into $\sqrt[3]{n}$ sections, each with $n^{2/3}$ rungs. We drop the first phone every $n^{2/3}$ rungs until it breaks. We then need to find the highest safe rung in the previous section of $n^{2/3}$ rungs. For this, we use our strategy with 2 phones, which performs $\Theta(\sqrt{n}$ pinePhone drops on n rungs. However, we perform this on a section with $n^{2/3}$ rungs, therefore performing $\Theta(\sqrt{n^{2/3}})$ pinePhone drops, which is $\Theta(\sqrt[3]{n})$. We drop the first phone $\sqrt[3]{n}$ times, the remaining phones $\Theta(\sqrt[3]{n})$ times, for a total of $\Theta(\sqrt[3]{n})$ pinePhone drops.

   (b) Now show that you can find the highest safe rung with 4 pinePhones with $\Theta(\sqrt[4]{n})$ pinePhone drops.

   Solution: Section the ladder into $\sqrt[4]{n}$ sections, each with $n^{3/4}$ rungs. We drop the first phone every $n^{3/4}$ rungs until it breaks. We then use our strategy for 3 phones on the previous section of $n^{3/4}$ rungs. We therefore drop the remaining 3 phones a total of $\Theta(\sqrt[3]{n^{3/4}}) = \Theta(\sqrt[4]{n})$ times. The total number of drops is therefore $\Theta(\sqrt[4]{n})$.

   (c) Suppose you continue your strategy to k pinePhones: Define the number of pinePhone drops as a recurrence in terms of n, the number of rungs on the ladder, and k, the number of pinePhones you are given. *Hint:* Make sure to give at least one base case. Do you need multiple base cases?

   Solution: Let $P(n, k)$ be the worst-case number of pinePhone drops on n rungs, with k pinePhones. Note that when we have 1 pinePhone, we use a linear strategy, performing n drops in the worst case. Following the pattern for 2, 3, and 4 pinePhones, when we increase the number of phones to k, we section the rungs into $\sqrt[k]{n}$ sections, each with $n^{(k-1)/k}$ rungs. We perform $\Theta(\sqrt[k]{n})$ drops on the boundaries of these sections, then perform our strategy for $k-1$ phones on the previous section with $n^{(k-1)/k}$ rungs. Then our recurrence is

$$P(n,k) = \begin{cases} n & \text{if } k = 1 \text{ or } n \leq 2, \\ P(n^{(k-1)/k}, k-1) + \Theta(\sqrt[k]{n}) & \text{if } n > 2 \text{ and } k > 1, \end{cases}$$

   which just has one base case.

Challenge: Solve this recurrence without using induction.