# CS 173: Discrete Structures, Fall 2009
# Homework 7 Solutions

This homework contains 5 problems worth a total of 42 regular points. It is due on Friday, October 23 at noon. Put your homework in the appropriate dropbox in the Siebel basement.

1. **Big-O proofs** [**8 points**]

   (a) Prove that $\frac{x^3+2x}{2x+1}$ is $O(x^2)$.
   [**Solution**]
   To satisfy the definition of $O(x^2)$, we need to find an appropriate choice of $c$ and $k$. Let's consider $x \geq 1000$ and reason by inequalities:

   $$\frac{x^3+2x}{2x+1} < \frac{x^3+2x}{2x} = \frac{1}{2}x^2 + 1 < \frac{1}{2}x^2 + x^2 = \frac{3}{2}x^2$$

   Since $\frac{x^3+2x}{2x+1} < \frac{3}{2}x^2$ when $x \geq 1000$, we can choose $k = 1000$ and $c = \frac{3}{2}$. (Note that there are an infinite number of correct choices of $k$ and $c$ that will work. This is just one approach to finding them.)

   (b) Use a proof by contradiction to show that $5^n$ is not $O(3^n)$.
   [**Solution**]
   Say that $5^n$ is $O(3^n)$. Then there exist $c$ and $k$ such that $5^n < c3^n$ for every $n \geq k$. This means:

   $$\begin{aligned} \log_3(5^n) &< \log_3(c3^n) \\ \log_3(5^n) &< \log_3(c) + \log_3(3^n) \\ \log_3(5^n) - \log_3(3^n) &< \log_3(c) \\ \log_3\left(\frac{5^n}{3}\right) &< \log_3(c) \\ n\log_3\left(\frac{5}{3}\right) &< \log_3(c) \\ n &< \frac{\log_3(c)}{\log_3(\frac{5}{3})} \\ n &< \log_{5/3}(c) \end{aligned}$$

   But this fails for $n$ greater than $\log_{5/3}(c)$, so we have a contradiction.

2. **Unrolling** [**8 points**]

   Use unrolling to find a closed form for the function $f : \mathbb{N} \to \mathbb{R}$ defined by

   - $f(0) = 0$

- $f(n) = 5f(n-1) + 1$ for all $n \geq 1$

Show at least three steps of unrolling, show the unrolling pattern compactly using a summation, then convert to a simple algebraic equation. Hint: you may need to consult a table of closed-forms for familiar summations, e.g. the one in Rosen section 2.4.

**[Solution]**

$$
\begin{aligned}
f(n) &= 5f(n-1) + 1 \\
&= 5(5f(n-2)+1) + 1 = 5^2 f(n-2) + 5 + 1 \\
&= 5^2(5f(n-3)+1) + 5 + 1 = 5^3 f(n-3) + 5^2 + 5 + 1 \\
&\quad \ldots \\
&= 5^i f(n-i) + \sum_{k=0}^{i-1} 5^k \\
&= 5^n f(0) + \sum_{k=0}^{n-1} 5^k \\
&= \sum_{k=0}^{n-1} 5^k = \frac{5^n - 1}{5 - 1} = \frac{5^n - 1}{4}
\end{aligned}
$$

3. **Recursion trees [8 points]**

Consider the following recurrence:

- $T(1) = d$
- $T(n) = T(n/3) + T(2n/3) + cn$

where $c$ and $d$ are constants. Let's analyze $T$ when the input $n$ is a power of 3.

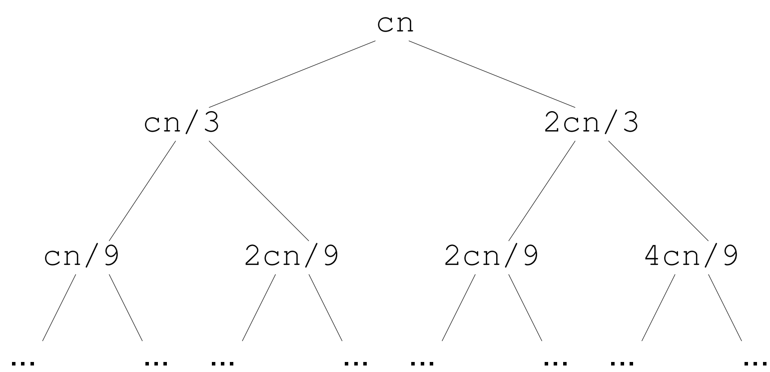
(a) Draw a recursion tree for $T$, in which each node shows the contribution of the non-recursive term $cn$.

(b) How high is this tree (exact height, not just big-O)?

(c) Use the tree to derive a closed-form solution for $T$, up to big-O.

Hint: see lecture 22 for examples. In part (c), "up to big-O" means that your solution should have the form "$T(n)$ is $O(xxx)$" and, therefore, you can ignore low-order terms and constant multipliers in your solution.

**[Solution]**




The bottom of the tree is ragged, i.e. not all branches will end at the same height. We know that each branch will end when the input to $T$ is less than or equal to 1. The deepest branch will be the rightmost one, wherein the input is decreased by a factor of $\frac{2}{3}$ at each step. Since the original input is $n$, we can find the number of levels in the tree $L$ by solving:

$$\begin{aligned} 1 &\geq n \cdot \left(\frac{2}{3}\right)^L \\ \left(\frac{3}{2}\right)^L &\geq n \\ L &\geq \log_{3/2}(n) \end{aligned}$$

$L$ will be the smallest integer that satisfies this, or $L = \lceil \log_{3/2}(n) \rceil$.

There's more than one possible way to define the height of a tree (e.g. number of levels, number of level transitions) and we haven't gotten to the official definition in lecture. So your solution might differ by one.

Now we need to find the running time. As we go down the tree from the top, each level contributes $cn$. This regularity will stop when the shortest branch (the leftmost one) ends. If we continue down from that level, each level will contribute (roughly) less than $cn$ as more and more branches terminate. Thus we can provide an upper bound on $T(n)$ by assuming that each of the $\log_{3/2}(n)$ levels contributes $cn$: $T(n) \leq cn \log_{3/2}(n)$ or $T(n) = O(n \log(n))$.

Note that we ignored the contribution of the termination points (or *leaves*) of the tree - where $T(1) = d$ is used. These will only occur $O(n)$ times, so their contribution will only be a "lower order" term in the closed form. Later in this course, we will become familiar with trees like this and will be able to show that the number of leaves is $O(n)$.

One way to understand why there are only $O(n)$ leaves is as follows. Each branch of the recursion tree terminates when the problem size $r$ is $\leq 1$. Since we are reducing the size by at most 3 at each level, then $r$ must be $\geq 1/3$ in these leaf nodes. (Otherwise this branch would have terminated one level earlier.) So we're dividing a total sum $cn$ at each level into chunks of size at least 1/3. So there can't be more than $3cn$ chunks, i.e. $O(n)$.

[**Another approach**] If we only consider the leftmost branch of the tree (the shortest one), we will also find a logarithmic number of levels ($k = \log_3(n)$). It's important to notice that most branches of the tree will continue deeper than that level. We can then argue that those deeper branches do not contribute more than $O(n\log(n))$ to the closed form.

[**Lower bound**] You only needed to produce an upper bound in your solution. However, notice that we can get a lower-bound on the complexity by summing up the node values for the filled part of the tree (before some branches start to terminate). So this recurrence is also $\Omega(n\log n)$, so therefore it's actually $\theta(n\log n)$.

4. **"Strong" induction [8 points]**

   Suppose that $f : \mathbb{Z}^+ \to \mathbb{Z}$ is defined by

   - $f(1) = 3$
   - $f(2) = 5$
   - $f(n) = 3f(n-1) - 2f(n-2)$ for all $n \geq 3$.

   Use strong induction to show that $f(n) = 2^n + 1$ for every positive integer $n$. Hint: you must use strong induction, because that's the main point of this problem.

   [**Solution**] We will prove this by induction on $n$.

   BASE CASES: When $n = 1$, then $f(1) = 3$ by definition and $2^1 + 1 = 3$, so the claim holds.

   When $n = 2$, then $f(2) = 5$ by definition and $2^2 + 1 = 5$, so the claim holds.

   INDUCTIVE HYPOTHESIS: Assume that $f(n) = 2^n + 1$ for all $n$ in the range $[1, k]$.

   INDUCTIVE STEP: Now we will show that the claim holds for $n = k + 1$. By definition:

   $$f(k+1) = 3f(k+1-1) - 2f(k+1-2) = 3f(k) - 2f(k-1)$$

   Applying the inductive hypothesis twice (once for the $n = k$ case and once for $n = k-1$):

   $$= 3(2^k + 1) - 2(2^{k-1} + 1) = 3 \cdot 2^k + 3 - 2^k - 2 = 2 \cdot 2^k + 1 = 2^{k+1} + 1$$

   This is what we wanted to show.

5. **Induction with an inequality [10 points]**

   Use induction to prove that the following equation holds for all integers $n \geq 2$:

   $$\sum_{k=n+1}^{2n} \frac{1}{k} \geq \frac{7}{12}$$

   [**Solution**] We will do induction on $n$.

   BASE CASE: When $n = 2$, $\displaystyle\sum_{k=2+1}^{2(2)} \frac{1}{k} = \frac{1}{3} + \frac{1}{4} = \frac{7}{12}$, so the claim holds.

   INDUCTIVE HYPOTHESIS: Assume that the claim holds for some $n = N$: $\displaystyle\sum_{k=N+1}^{2N} \frac{1}{k} \geq \frac{7}{12}$.

Inductive Step: We will show the claim holds for $n = N+1$. That is, $\displaystyle\sum_{k=N+2}^{2N+2} \frac{1}{k} \geq \frac{7}{12}$.

Consider the summation:

$$\sum_{k=N+2}^{2N+2} \frac{1}{k} = \left(\sum_{k=N+1}^{2N+2} \frac{1}{k}\right) - \frac{1}{N+1} = \left(\sum_{k=N+1}^{2N} \frac{1}{k}\right) - \frac{1}{N+1} + \frac{1}{2N+1} + \frac{1}{2N+2}$$

$$= \left(\sum_{k=N+1}^{2N} \frac{1}{k}\right) + \frac{1}{2N+1} - \frac{1}{2N+2}$$

Now apply the inductive hypothesis and note that for positive $N$, $\frac{1}{2N+1}$ is greater than $\frac{1}{2N+2}$ and both are positive, so $\frac{1}{2N+1} - \frac{1}{2N+2}$ is greater than 0:

$$\left(\sum_{k=N+1}^{2N} \frac{1}{k}\right) + \frac{1}{2N+1} - \frac{1}{2N+2} \geq \frac{7}{12} + \frac{1}{2N+1} - \frac{1}{2N+2} > \frac{7}{12}$$

This is what we wanted to show.