CS 31: Algorithms (Spring 2019): Lecture 3

Date: 28th March, 2019

Topic: Divide and Conquer – Merge Sort, Recurrences, Counting Inversions

Disclaimer: These notes have not gone through scrutiny and in all probability contain errors. Please email errors to deeparnab@dartmouth.edu.

---

1 Proof of the Master Theorem

Theorem 1. Consider the following recurrence:

$$T(n) \leq a \cdot T(\lceil n/b \rceil) + \Theta(n^d)$$

where a, b, d are non-negative integers. Then, the solution to the above is given by

$$T(n) = \begin{cases} O(n^d) & \text{if} \quad a < b^d \\ O(n^d \log n) & \text{if} \quad a = b^d \\ O(n^{\log_b a}) & \text{if} \quad a > b^d \end{cases}$$

Proof. As before, we replace the $\Theta(n^d)$ by $\leq C \cdot n^d$ for some constant $C > 0$. We also replace the $\lceil n/b \rceil$ by the simpler n/b and worry about how to handle these later. Once again, you should try drawing the recursion tree and see how the kitty grows. Here we give the "opening the brackets and consolidating" method.

$$\begin{aligned} T(n) &\leq aT(n/b) + C \cdot n^d \\ &\leq a\left(aT(n/b^2) + C \cdot (n/b)^d\right) + C \cdot n^d \\ &= a^2T(n/b^2) + Cn^d \cdot \left(1 + \frac{a}{b^d}\right) \\ &\leq a^2\left(aT(n/b^3) + C \cdot \left(n/b^2\right)^d\right) + Cn^d \cdot \left(1 + \frac{a}{b^d}\right) \\ &= a^3T(n/b^3) + Cn^d \cdot \left(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2\right) \\ &\vdots \\ &\leq a^kT(n/b^k) + Cn^d \cdot \left(1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 + \cdots + \left(\frac{a}{b^d}\right)^{k-1}\right) \qquad (1) \\ & \qquad (2) \end{aligned}$$

To finish the proof, we need to argue about the Geometric Sum in (1). Note that when $a = b^d$, the geometric sum evaluates to k. When $a < b^d$, then if we denote $1 > \rho = a/b^d$, the Geometric Sum evaluates to $\leq \frac{1}{1-\rho} = \Theta(1)$ since a, b, d are constants. When $a > b^d$, then if we denote $1 < \rho = a/b^d$, the Geometric Sum evaluates to $\frac{\rho^k - 1}{\rho - 1}$. In sum, we get for any k,

$$T(n) = a^k T(n/b^k) + Cn^d \cdot \begin{cases} \Theta(k) & \text{if } a = b^d \\ \Theta(1) & \text{if } a < b^d \\ \Theta((a/b^d)^k) & \text{if } a > b^d \end{cases}$$

To finish the proof, we set $k = \log_b n$ in which case the first term $a^k T(n/b^k)$ evaluates to $\Theta(a^{\log_b n}) = \Theta(n^{\log_b a})$. When $a \leq b^d$, we get $n^{\log_b a} \leq n^d$. Thus in Case 1 and Case 2 above, we see that the Master theorem follows. The third case of the above cases evaluates to

$$\left(\frac{a}{b^d}\right)^k = \left(b^{\log_b(a/b^d)}\right)^k = \left(b^{\log_b n}\right)^{\log_b(a/b^d)} = n^{(\log_b a - d)} = n^{\log_b a}/n^d$$

Therefore, in the third case the second summand evaluates to $\Theta(n^{\log_b a})$. □

2 Taking care of ceilings and floors.

Let's come back to that icky detail of "forgetting" the ceilings and floors. Let's recall the true mergesort recurrence

$$T(n) \leq T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + a \cdot n \quad \leq \quad 2T(n/2 + 1) + a \cdot n \tag{3}$$

where we have (a) used the monotonicity of $T(\cdot)$ and (b) used $\lfloor x \rfloor \leq \lceil x \rceil < x + 1$.
Now consider the function $S(n) = T(n+2)$.

Claim 1. $S(1) = \Theta(1)$ and for all $n > 1$, $S(n) \leq 2S(n/2) + \Theta(n)$

Proof. $S(1) = T(3) = \Theta(1)$; there is nothing to prove here. To see the recurrence, observe

$$\begin{aligned} S(n) = T(n+2) \quad &\leq \quad 2T\left(\frac{n+2}{2} + 1\right) + a(n+2) \\ &= \quad 2T\left(\frac{n}{2} + 2\right) + (a \cdot n + 2a) \\ &= \quad 2S(n/2) + \Theta(n) \end{aligned}$$

□

Since we know from the previous argument that $S(n) = \Theta(n \log n)$, we get $T(n) = \Theta((n-2)\log(n-2)) = \Theta(n \log n)$. Such an idea takes care of floors and ceilings in many arguments.

Exercise:

- *Complete the proof of the Master Theorem with ceilings using the transformation* $S(n) = T(n+b)$.
- *Solve the recurrence* $T(n) \leq T(\lfloor n/3 \rfloor) + T(\lceil 2n/3 \rceil) + \Theta(n)$.
- *Solve the recurrence* $T(n) \leq \sqrt{n} \cdot T(\lceil \sqrt{n}\, \rceil) + \Theta(n)$. ☕☕

Hint: Think of the 'transformation' : $S(n) = (1 + \frac{4}{n}) \cdot T(n+4)$