CSE 417: Algorithms and Computational Complexity **Winter 2012**

Sample Exercises on Recurrences

In this handout, we give few examples of recurrences and how to solve them. We start by stating Master's theorem:

Theorem 1 (Master's Theorem) . *Let a and b be positive constants and let $T(n) = aT(n/b) + cn^k$ for $n > b$ then*

- *if $a > b^k$ then $T(n)$ is $\Theta(n^{\log_b a})$*
- *if $a < b^k$ then $T(n)$ is $\Theta(n^k)$*
- *if $a = b^k$ then $T(n)$ is $\Theta(n^k \log n)$*

Master's theorem works even if we are using $\lceil \frac{n}{b} \rceil$ instead of $\frac{n}{b}$.

Exercises

Solve the following recurrence to get the best asymptotic bounds you can on $T(n)$ in each case using O( ) notation.

1. $T(n) = T(\frac{n}{4\log_2 n}) + 2n$ for $n > 1$ and $T(1) = 1$. You can assume that all numbers is rounded down to the nearest integer.

   Solution We will provide an upper and a lower bound to show that the best asymptotic bound for the recurrence is $\Theta(n)$. A lower bound of $\Omega(n)$ follows from the $2n$ term in $T(n)$.

   An upper bound can be reached by observing that $4\log n > 4$ for $n > 1$, so $T(\frac{n}{4(\log n)}) \leq T(\frac{n}{4})$ and we have

   $$T(n) \leq T\left(\frac{n}{4}\right) + 2n$$

   which, by Master's theorem, is $O(n)$. It follows that $T(n) \in \Theta(n)$.

2. BFPRT algorithm for median finding: $T(n) \leq cn + T(\frac{n}{5}) + T(\frac{3n}{4})$ and $T(1) = 1$, where $c \geq 1$. You can assume that everything is rounded down to the nearest integer.

   Solution We prove by induction on n that $T(n) \leq 20cn = O(n)$.

   (a) Base case: $T(1) = 1 \leq 20c$, done.

   (b) Inductive hypothesis: Let $k \geq 2$. Assume $T(i) \leq 20ci$ for all $i \leq k - 1$.

(c) Inductive step: We prove the statement true for k.

$$\begin{aligned} T(k) &\leq ck + T(\frac{k}{5}) + T(\frac{3k}{4}) \\ &\leq ck + 20c\frac{k}{5} + 20c\frac{3k}{4} \\ &= ck\left(1 + 20\left(\frac{19}{20}\right)\right) \\ &= ck(1+19) \\ &= 20ck \end{aligned}$$

Notice that although we have two subproblems here, we have an $O(n)$ running time since the size of the two subproblems combined is strictly less than n.

3. Polynomial Multiplication: $T(n) = 3T(\lceil\frac{n}{2}\rceil) + cn$ and $T(1) = 4$.

   Solution Since $3 > 2^1$, by the Master's theorem, we have $T(n) = \Theta(n^{\log_2 3})$.

4. Matrix Multiplication: $T(n) = 7T(\lceil\frac{n}{2}\rceil) + cn^2$ and $T(1) = 1$.

   Solution Since $7 > 2^2$, by the Master's theorem, we have $T(n) = \Theta(n^{\log_2 7})$.

5. $T(n) \leq T\left(\lceil\frac{n}{2}\rceil\right) + T\left(\lfloor\frac{n}{2}\rfloor\right) + cn\log n$, for $n \geq 4$ and $T(2) = 2$.

   Solution We prove the by induction on n that $T(n) \leq cn\log^2 n) = O(n\log^2 n)$

   (a) Base case: $T(2) = 2$ and $2\log^2 2 = 2$, done.

   (b) Inductive hypothesis: Let $k \geq 2$. Assume $T(i) \leq ci\log^2 i$ for all $i \leq k-1$.

   (c) Inductive step: We prove the statement true for k.

$$\begin{aligned} T(k) &\leq T\left(\lceil\frac{k}{2}\rceil\right) + T\left(\lfloor\frac{n}{2}\rfloor\right) + ck\log k & (1) \\ &\leq c\left(\lceil\frac{k}{2}\rceil\log^2\lceil\frac{k}{2}\rceil\right) + c\left(\lfloor\frac{k}{2}\rfloor\log^2\lfloor\frac{k}{2}\rfloor\right) + ck\log k & (2) \\ &\leq c\left(\lceil\frac{k}{2}\rceil\log^2\lceil\frac{k}{2}\rceil\right) + c\left(\lfloor\frac{k}{2}\rfloor\log^2\lceil\frac{k}{2}\rceil\right) + ck\log k & (3) \\ &= ck\left(\log^2\lceil\frac{k}{2}\rceil\right) + ck\log k & (4) \\ &\leq ck\log k\log\lceil\frac{k}{2}\rceil + ck\log k & (5) \\ &\leq ck\log k(\log k - 1) + ck\log k & (6) \\ &= ck\log^2 k & (7) \end{aligned}$$

   where (3) follows from the fact that $\lfloor\frac{k}{2}\rfloor \leq \lceil\frac{k}{2}\rceil$, (4) from the fact that $\lfloor\frac{k}{2}\rfloor + \lceil\frac{k}{2}\rceil = k$ and (6) from the fact that $\log\lceil\frac{k}{2}\rceil \leq \log k - 1$.