COMP108 Algorithmic Foundations

Tutorial 6 (Suggested solution and Feedback) w/c 13th March 2017

1. *Feedback: Common mistakes include mixing up* $T(\frac{n}{4})$ *and* $\frac{n}{4}$*,* $T(\frac{n}{4})$ *means the value of the function* $T(\,)$ *when the parameter is* $\frac{n}{4}$*, it is not the same as* $\frac{n}{4}$*.*

   Given the recurrence:

$$T(n) = \begin{cases} 3 & \text{if } n = 1 \\ 4 \times T(\frac{n}{4}) + 3 & \text{if } n > 1 \end{cases}$$

   Prove that $T(n)$ is $O(n)$ by the *substitution method*, i.e., use mathematical induction.

   We are going to prove that $T(n) = 4 \times n - 1$ for all $n \geq 1$.

   Base case:

   When $n = 1$, LHS $= T(1) = 3$, RHS $= 4 \times 1 - 1 = 3$.

   Therefore, LHS = RHS

   Induction hypothesis: Assume that the property holds for all integers $n' < n$, i.e., assume

$$T\left(\frac{n}{4}\right) = 4 \times \frac{n}{4} - 1$$

   Induction step:

   We want to prove $\boldsymbol{T(n) = 4 \times n - 1}$.

   [The induction step can be proved by first using the recurrence to express $T(n)$ in terms of $T(\frac{n}{4})$, and then use the hypothesis.]

$$\begin{aligned}
\text{LHS} = T(n) &= 4 \times T\left(\frac{n}{4}\right) + 3 && \leftarrow \text{ use the recurrence} \\
&= 4 \times \left(4 \times \frac{n}{4} - 1\right) + 3 && \leftarrow \text{ use induction hypothesis} \\
&= 4 \times (n-1) + 3 && \leftarrow \text{arithmetic} \\
&= 4 \times n - 1
\end{aligned}$$

   Therefore, LHS. = RHS and the property holds for n.

   Conclusion: $T(n) = 4 \times n - 1$ for all positive integers n and therefore, $T(n)$ is $O(n)$.

2. Given the recurrence

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4 \times T(\frac{n}{4}) + n & \text{if } n > 1 \end{cases}$$

Prove that $T(n)$ is $O(n \log n)$ by the *substitution method*, i.e., use mathematical induction.

We are going to prove that $T(n) \leq 4 \times n \times \log n$ for all $n \geq 4$.

Base case:

When $n = 4$,
L.H.S. $= T(4) = 4 \times T(1) + 4 = 4 \times 1 + 4 = 8$
R.H.S. $= 4 \times 4 \times \log 4 = 32$

Therefore, L.H.S. $\leq$ R.H.S.

Induction hypothesis: Assume that the property holds for all integers $n' < n$, i.e., assume

$$T\left(\frac{n}{4}\right) \leq 4 \times \frac{n}{4} \times \log(\frac{n}{4})$$

Induction step:

We want to prove $\boldsymbol{T(n) \leq 4 \times n \times \log n}$.

$$\begin{aligned}
\text{L.H.S.} = T(n) &= 4 \times T\left(\frac{n}{4}\right) + n && \leftarrow \text{ use the recurrence} \\
&\leq 4 \times (4 \times \frac{n}{4} \times \log(\frac{n}{4})) + n && \leftarrow \text{ use induction hypothesis} \\
&= 4 \times (n \times \log(\frac{n}{4})) + n && \leftarrow \text{arithmetic} \\
&= 4 \times (n \times (\log n - \log 4)) + n && \leftarrow \log(\tfrac{x}{y}) = \log x - \log y \\
&= 4 \times (n \times (\log n - 2)) + n \\
&= 4 \times (n \times \log n - 2n) + n \\
&= 4 \times n \times \log n - 8n + n \\
&< 4 \times n \times \log n \\
&= R.H.S.
\end{aligned}$$

Therefore, L.H.S. $\leq$ R.H.S. and the property holds for n.

Conclusion: $T(n) = 4 \times n \times \log n$ for all positive integers $n \geq 4$ and therefore, $T(n)$ is $O(n \log n)$.

3. If there is only one value in the array, return this value.

   Otherwise, divide the array into two halves, and **recursively** find the product of the values in each half.

   Suppose the two values we obtain are $product_1$ and $product_2$, respectively.

   Return $product_1 * product_2$.

   Pseudo code:

   Algorithm RecProduct($A[1..n]$)
   if $n == 1$ **then** // only one value
   return $A[n]$
   else
   begin Copy $A[1..\frac{n}{2}]$ to $B[\,]$
   Copy $A[(\frac{n}{2}+1)..n]$ to $C[\,]$
   $product1 = RecProduct(B[1..\frac{n}{2}])$
   $product2 = RecProduct(C[1..\frac{n}{2}])$
   return $product1 * product2$
   end

   Note: we invoke this algorithm by calling RecProduct($A[1..n]$).