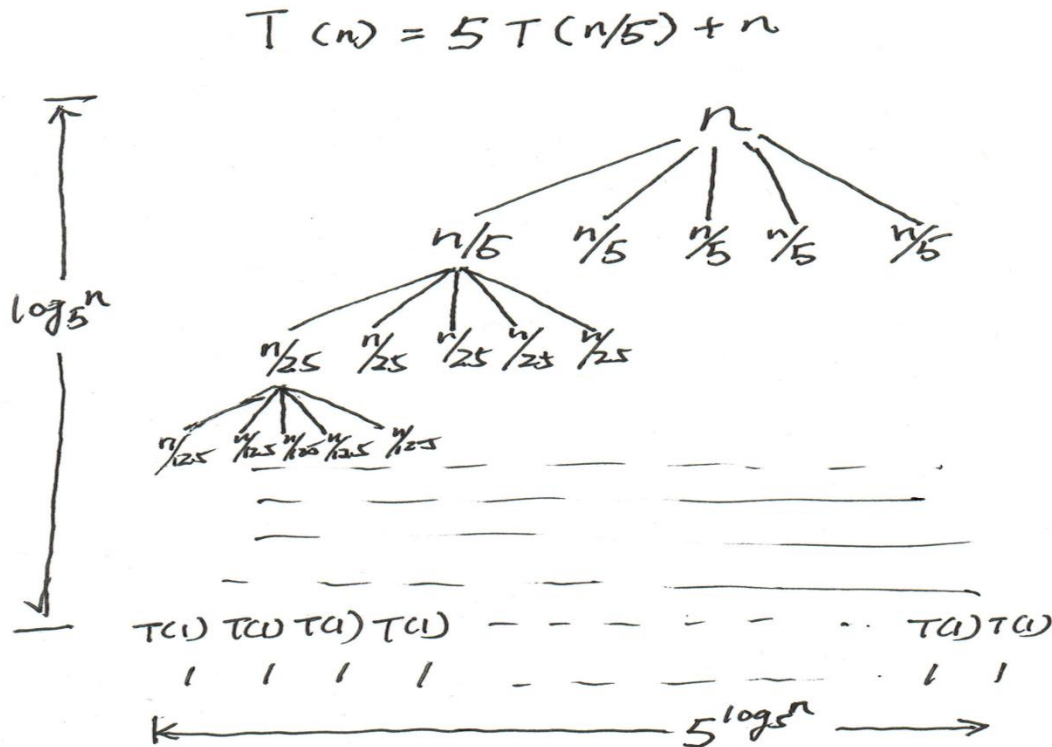


The running time of an algorithm can be described by the equation

$T(n) = 5T(n/5) + n$. Draw a Recursion Tree to find the exact function $T(n)$ and measure the run time complexity big O for $T(n)$.

Answer:



$$\begin{aligned}
 n & \dots \text{level } 1 = n \\
 5 \left(\frac{n}{5} \right) & \dots \text{level } 2 = n \\
 5^2 \left(\frac{n}{5^2} \right) & \dots \text{level } 3 = n \\
 5^3 \left(\frac{n}{5^3} \right) & \dots \text{level } 4 = n \\
 & \vdots \\
 5^i \left(\frac{n}{5^i} \right) & \dots \text{level } i = n \\
 1 \times n & = n
 \end{aligned}$$

$$T(n) = n \log_5 n + n$$

Use Master Theorem:

$$f(n) = O(n^{\log_b a}) = O(n), \quad T(n) = \Theta(n^{\log_b a} \lg n) = O(n \lg n).$$