3. (25%)

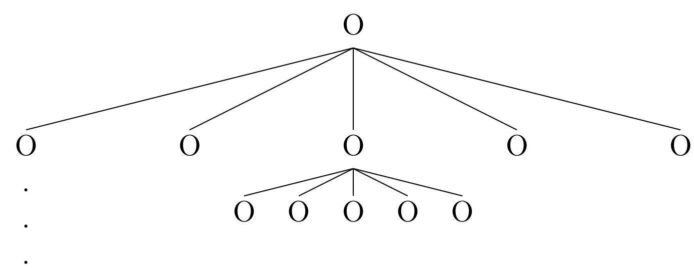
(a) The recurrence relation for $T(n)$ of the modified algorithm is:

$$T(n) = \begin{cases} C_0 & \text{if } x = 1 \\ 5T(\frac{n}{5}) + cn & \text{otherwise} \end{cases}$$

This is the same as:

$$T(n) = c_0 n + 5 \sum_{k=0}^{\log_5(\frac{n}{5})-1} cn$$

(b) Recurrence Tree:



(c) We wish to show that $T(n) = \Theta(n\log(n))$

$$\begin{aligned} T(n) &= c_0 n + 5 \sum_{k=0}^{\log_5(\frac{n}{5})-1} cn \\ &= n(c_0 + 5c\log_5\left(\frac{n}{5}\right) - c) \\ &= n(c_0 + 5c\log_5(n) - \log_5(5) - c) \\ &= n(c_0 + 5c\log_5(n) - 1 - c) \\ &= n(c_1 + 5c\log_5(n)) \\ &= n(c_1 + 5c\left(\frac{\log_2(n)}{\log_2(5)}\right)) \\ &= n(c_1 + 5c_2\log_2(n)) \\ &= nc_1 + c_3 n\log_2(n) \\ &= \Theta(n\log(n)) \end{aligned}$$

(d) By dividing the problem into m subproblems, we divide n into m pieces at each level, and repeat each subproblem m times. This results in a different logarithmic base and leading coefficient. Because of the change of base, this changes the constant coefficient multiplier, which doesn't alter the asymptotic result.