3. (25%)

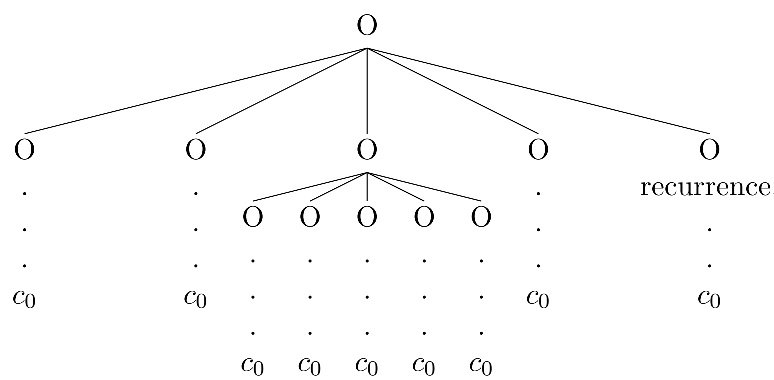
(a) The recurrence relation for $T(n)$ of the modified algorithm is:

$$T(n) = \begin{cases} c_0 & \text{if } x = 1 \\ 5T(\frac{n}{5}) + cn & \text{otherwise} \end{cases}$$

This is the same as:

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

(b) Recurrence Tree:



where the work per node is $\frac{cn}{5^k}$ where k is the level of the node. The work per level can then be generalized to $5^k(\frac{cn}{5^k}) = cn$.

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

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

(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.