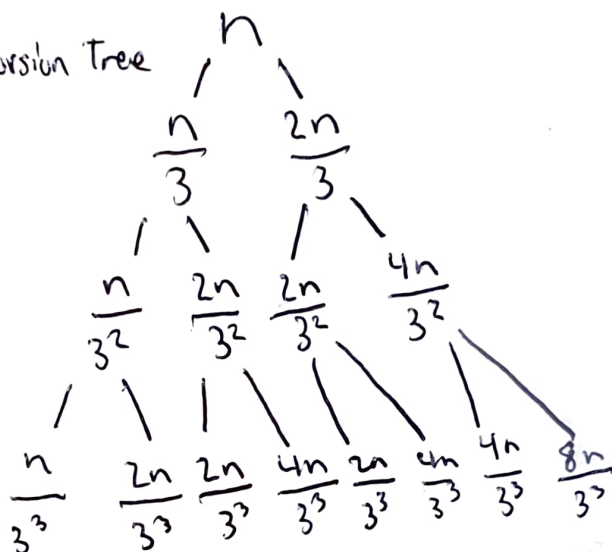


Problem 2

$$1. T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n)$$

$$= T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

Solve by Recursion Tree



$$\frac{2n}{3} \Rightarrow \frac{4n}{3^2} \Rightarrow \frac{8n}{3^3}$$

$$\frac{n}{(\frac{3}{2})} \Rightarrow \frac{n}{(\frac{3}{2})^2} \Rightarrow \frac{n}{(\frac{3}{2})^3}$$

$$= \frac{n}{(\frac{3}{2})^k}$$

$$c \cdot n \cdot \log_{\frac{3}{2}} n$$

$$k = \log_{\frac{3}{2}} n$$

$$= O(n \log_{\frac{3}{2}} n)$$

$$2. T(n) = 7T\left(\frac{n}{2}\right) + n^2$$

Solve by Master's Theorem

$$a = 7 \quad b = 2 \quad f(n) = n^2$$

$$O(n^{\log_2 7 - \epsilon}) = O(n^{\log_2 7 - \epsilon})$$

$$T(n) = O(n^{2-\epsilon}) \text{ for } \epsilon = 0.807$$

$$\therefore T(n) = O(n^2)$$