

4.1

a) are in the zip file.

c)

Best - case, is dependent on the time complexity of insertion sort. For different, k ^{increasing} values, it requires less time. When k becomes large insertion sort is applied, which has $O(n)$.

Average case is dependent on both merge & insertion sort, and on the k .

$$So, O(n) = \left[\frac{n}{k} \log \frac{n}{k} + \left(\frac{n}{k} \right)^2 \right]$$

Worst case is dependent on insertion sort because as array size gets too large $n^2 > n \log n$, hence $O(n) = n^2$.

4.2

a) $T(n) = 36 T\left(\frac{n}{6}\right) + 2n$

$a = 36, b = 6$

$n^{\log_b a} = n^2$

$f(n) = 2n$

$f(n) = O(n^{2-\epsilon})$ for $\epsilon = 1$

Since, $\epsilon > 0, T(n) = \Theta(n^2)$

b) $T(n) = 5 T\left(\frac{n}{3}\right) + 17n^{1.2}$

$a = 5, b = 3$

$n^{\log_b a} = n^{1.46}$
 $f(n) = 17n^{1.2}$

$f(n) = O(n^{\log_b a - \epsilon})$

$\therefore T(n) = \Theta(n^{1.46})$

$\log_3 5 - \epsilon = 1.2$

$\epsilon = 0.26$

c) $T(n) = 12 T\left(\frac{n}{2}\right) + n^2 \log n$

$a = 12, b = 2, f(n) = n^2 \log n$

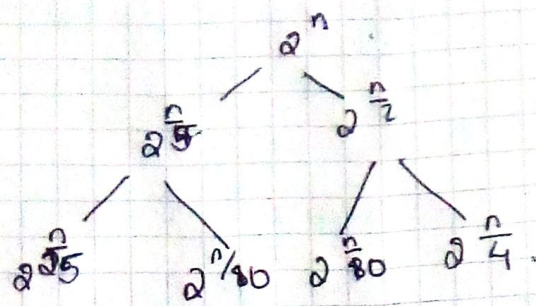
$n^{\log_2 12} = 8.58$

Next, case I:

$n^{8.58} > n^2 \log n$ will be greater for some value.

$\therefore T(n) = \Theta(n^{8.58})$

a) $T(n) = 3T\left(\frac{n}{5}\right) + T\left(\frac{n}{7}\right) + 2^n$



2^n
 $2^{n/5} + 2^{n/7}$

$2^{n/25} + 2^{n/35} + 2^{n/35} + 2^{n/49}$

$2^n + 2^{n/5} + 2^{n/7} + 2^{n/25} + 2^{n/35} + 2^{n/49} + \dots$
 $\Rightarrow 2^n + (2^n)^{1/5} + (2^n)^{1/7} + (2^n)^{1/25} + (2^n)^{1/35} + (2^n)^{1/49} + \dots$

$\Theta(1)$

Lower bound:

$2^n, 2^{n/5}, 2^{n/25}, \dots, (2^n)^{1/5^i}$

$3 = (2^n)^{1/5^i}$

$\Rightarrow i = \log_5 \left(n \cdot \frac{\ln 2}{\ln 3} \right)$

Next,
 $cn \left(1 + \log_5 n \cdot \frac{\ln 2}{\ln 3} \right) \Rightarrow \Omega(n) = n \log_5 n$

Since bounds only differ by constant value

$\Theta(n) = n \log n$

Upper bound:

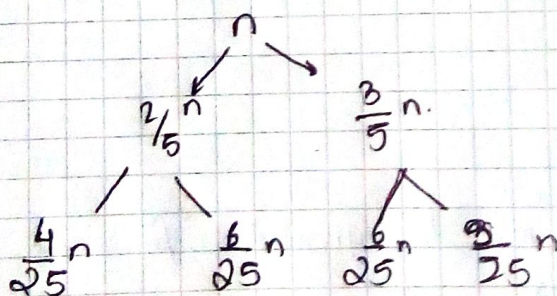
$(2^n)^{1/2} = 3$

$\Rightarrow i = \log_2 \left(n \cdot \frac{\ln 2}{\ln 3} \right)$

Next,
 $cn \left(1 + \log_2 n \cdot \frac{\ln 2}{\ln 3} \right)$

$\Theta(n) = n \log_2 n$

b) $T(n) = T\left(\frac{2n}{5}\right) + T\left(\frac{3n}{5}\right) + \Theta(n)$



$n + n + n$

Lower bound: $\frac{2}{5}n, \frac{4}{25}n, \dots$ (left path)

$n, \left(\frac{2}{5}\right)n, \left(\frac{2}{5}\right)^2 n, \dots, 1 = n \left(\frac{2}{5}\right)^i$

$1 = n \left(\frac{2}{5}\right)^i$

$n = \left(\frac{5}{2}\right)^i \Rightarrow i = \log_{5/2} n$

upper bound: (right path)

$$n \left(\frac{3}{5} \right)^i = 1.$$

$$i = \log_{\frac{5}{3}} n.$$

Since, the so, there are $1 + \log_{\frac{5}{3}} n$ levels. (1 is for the top).
and each level will at most cn time.

$$\begin{aligned} & \underbrace{cn + cn + \dots + cn}_{\substack{\downarrow \\ 1 + \log_{\frac{5}{3}} n \text{ times}}} \\ &= cn(1 + \log_{\frac{5}{3}} n) \\ &= cn + cn \frac{\log n}{\log \frac{5}{3}}. \end{aligned}$$

which is ignoring lower order terms (cn) and constants
 $\left(\frac{c}{\log \frac{5}{3}} \right)$ we get $\Theta(\text{upper bound}) = n \log_{\frac{5}{3}} n.$

Same is applied for lower bound $= n \log_{\frac{5}{2}} n.$

Since, bounds only differ by constant value.

$$\Theta(n) = n \log n.$$