Homework 2.2

a) $36T(n/6) + 2n$ <u>Use Master Method</u>

$a = 36$

$b = 6$ $\log_b a = \log_6 36$

$f(n) = 2n$ $n^{\log_6 36} = n^2$

n^2 is significantly greater than n for larger n. Hence, it is the first case of Master Theorem. Therefore, the solution to this recurrence is $\Theta(n^{\log_b a})$

So $T(n) = \Theta(n^2)$

b) $5T(n/3) + 17n^{1.2}$ <u>Use Master Method</u>

$a = 5$

$b = 3$ $\log_b a = \log_3 5$

$f(n) = 17n^{1.2}$ $n^{\log_3 5} \approx n^{1.464}$

$n^{1.464}$ is polynomially greater than $17n^{1.2}$. Hence it is the first case of Master Theorem. Therefore, the solution to this recurrence is $\Theta(n^{\log_b a})$

So $T(n) = \Theta(n^{\log_3 5}) \approx \Theta(n^{1.464})$

c) $12T(n/2) + n^2 \lg n$

$a = 12$

$b = 2$ $\log_b a = \log_2 12$

$f(n) = n^2 \lg n$ $n^{\log_2 12} \approx n^{3.58}$

$n^{3.58}$ is significantly greater than $n^2 \lg n$, because $n^{3.585} > n^2$ and $\Theta(n) \gg \Theta(\log n)$. Hence it is the first case of Master Theorem. Therefore, the solution is $\Theta(n^{\log_b a})$

$$\text{So} \quad T(n) = \Theta(n^{\log_2 12})$$

d) $T(n) = 3T(n/5) + T(n/2) + 2^n$

We need to find the lower bound:

$$T(n) \geq 4T(n/5) + 2^n$$

Solve for:

$$T(n) = 4T(n/5) + 2^n$$

By Master Method:

$a = 4$

$b = 5$ $\quad \log_b a = \log_5 4$

$f(n) = 2^n$ $\quad n^{\log_5 4} = n^{0.86}$

$n^{0.86}$ is polynomially smaller than 2^n. Hence, it is the third case of Master Theorem. Therefore the solution is $\Theta(f(n))$

$$\text{So} \quad T(n) = \Theta(2^n)$$

We check the Regularity Condition now:

$$af(n/b) \leq cf(n), \quad \text{where } c < 1$$

$$4 \cdot 2^{n/5} \leq c \cdot 2^n \Rightarrow 2^{-4n/5} \leq c/4$$

- This is true as n grows, $2^{-4/5} \rightarrow 0$ in the end

We need to find the upper bound:

$$T(n) \leq 4T(n/2) + 2^n$$

Solve for:

$$T(n) = 4T(n/2) + 2^n$$

By Master Method:

$a = 4$

$b = 2$ $\quad \log_b a = \log_2 4$

$f(n) = 2^n$ $\quad n^{\log_2 4} = n^2$

n^2 is polynomially smaller than 2^n. Hence, it is the third case of Master Theorem. Therefore the solution is $\Theta(f(n))$

So $T(n) = \Theta(2^n)$

We check the Regularity Condition now:

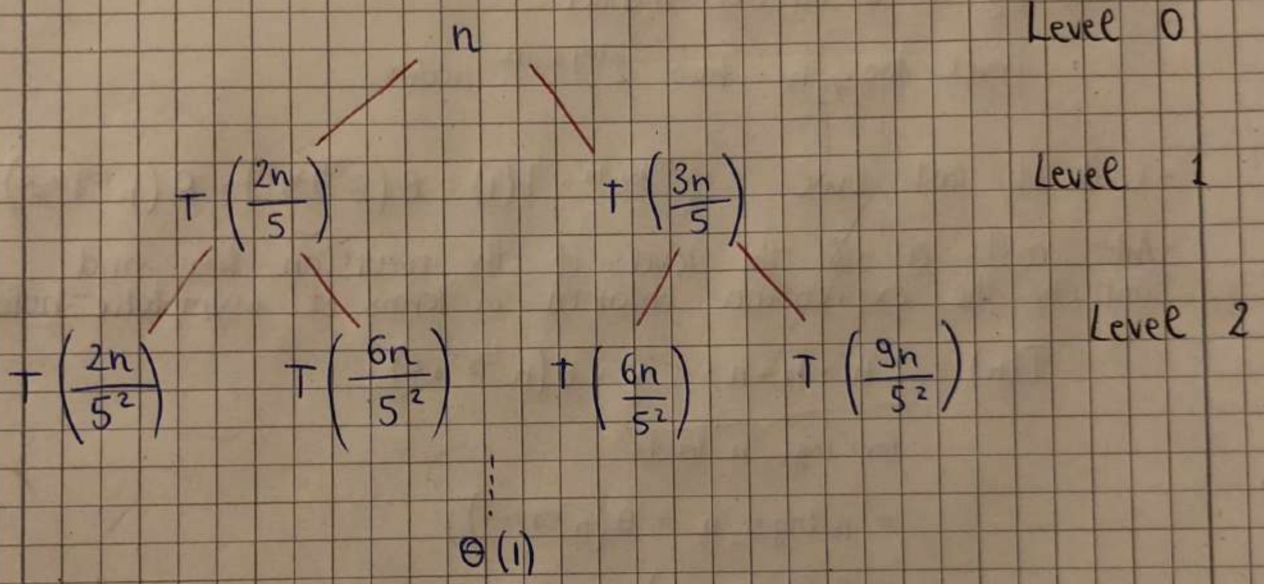
$$af(n/b) \le cf(n), \text{ where } c < 1$$

$$4 \cdot 2^{n/2} \le c \cdot 2^n \Rightarrow 2^{-n/2} \le c/4$$

It is true as n increases, $2^{-n/2} \to 0$ in the end

$\Rightarrow$ Since the lower and upper bounds for $T(n)$ are equal, the final result: $T(n) = \Theta(2^n)$

e) $T(n) = T(2n/5) + T(3n/5) + \Theta(n)$ Use Recursion Tree



- Cost of level-0 $= n$
- Cost of level-1 $= \frac{2n}{5} + \frac{3n}{5} = n$
- Cost of level-2 $= \frac{2n}{5^2} + \frac{6n}{5^2} + \frac{6n}{5^2} + \frac{9n}{5^2} = n$

Determine the total number of levels in the recursion tree. We will consider the rightmost sub tree as it goes to the deepest level:

- Size of sub-problem at level-0 : $(3/5)^0 n$
- Size of sub-problem at level-1 : $(3/5)^1 n$
- Size of subproblem at level-2 : $(3/5)^2 n$

Continuing in a similar manner, we have:

- size of sub-problem at level-i $= (3/5)^i n$

Suppose at level-x (last level), size of sub-problem becomes 1. Then:

$$(3/5)^x n = 1$$

$$(3/5)^x = 1/n$$

$$x \log(3/5) = \log(1/n)$$

$$x = \log_{5/3} n$$

Total number of levels in the recursion tree = $\log_{5/3} n + 1$

Determine number of nodes in the last level :

- Level - 0 has 2^0 nodes = 1 node
- Level - 1 has 2^1 nodes = 2 nodes
- Level - 2 has 2^2 nodes = 4 nodes

Continuing in a similar manner :

Level - $\log_{5/3} n$ has $2^{\log_{5/3} n}$ nodes

- Cost of last level = $2^{\log_{5/3} n} \cdot T(1) = \theta(2^{\log_{5/3} n}) = \theta(n^{\log_{5/3} 2})$
- Add costs of all the levels of the recursion tree and simplify the expression obtained in terms of asymptotic notation :

$$T(n) = \underbrace{\{n + n + n + \dots\}}_{\text{for } \log_{5/3} n \text{ levels}} + \theta(n^{\log_{5/3} 2})$$

$$= n \log_{5/3} n + \theta(n^{\log_{5/3} 2})$$

$$= \theta(n \log_{5/3} n)$$