Merge sort $O(N[k + \log(N/k)])$

a) ✓

1, b) ✓

c) Worst case → Increasing the value of k, means that insertionsort will be applied to a bigger (in size) subarray, so more shifting will take place as elements are in reverse ⇒ the execution time increases with the increment of k.

* The result is more visible with bigger arrays.

Average case → Increasing the value of k, also increases the time of compilation in general, because by exactly how much depends on the arrangment of the elements inside the array.

Best case → The compilation time decreases with the increment of k as it takes more time to divide using mergesort compared to insertionsort. For best case insertion sort $\in O(n)$

mergesort $\in O(n \log n)$

d) For worst case, k should be ~~small~~ small. Insertionsort takes more time as more comparisons take place, while mergesort divides all the reverse-sorted elements and compares in smaller cunks of arrays.

For average case, k depends on the alternation on the elements inside the array (in which part they are sorted), but it doesn't make much difference.

For best case: k should be equal to the size of the array as in insertionsort only the for loop would be executed $\Rightarrow \in O(n)$, while in merge sort is $\in O(n \log n)$

2, a) $T(n) = 36T(n/6) + 2n$

Using master theorem $a = 36$ $b = 6$ $f(n) = \Theta(n)$

$n^{\log_b a} = n^2 > f(n)$

Thus $\boxed{T(n) = \Theta(n^2)}$

Homework 4

4.2

b) $T(n) = 5T(n/3) + 17n^{1,2}$ $\qquad f(n) = n^{1,2}$

$n^{\log_b a} = n^{\log_3 5} = n^{1,46} > f(n)$

Thus $T(n) \in \Theta(n^{\log_3 5})$

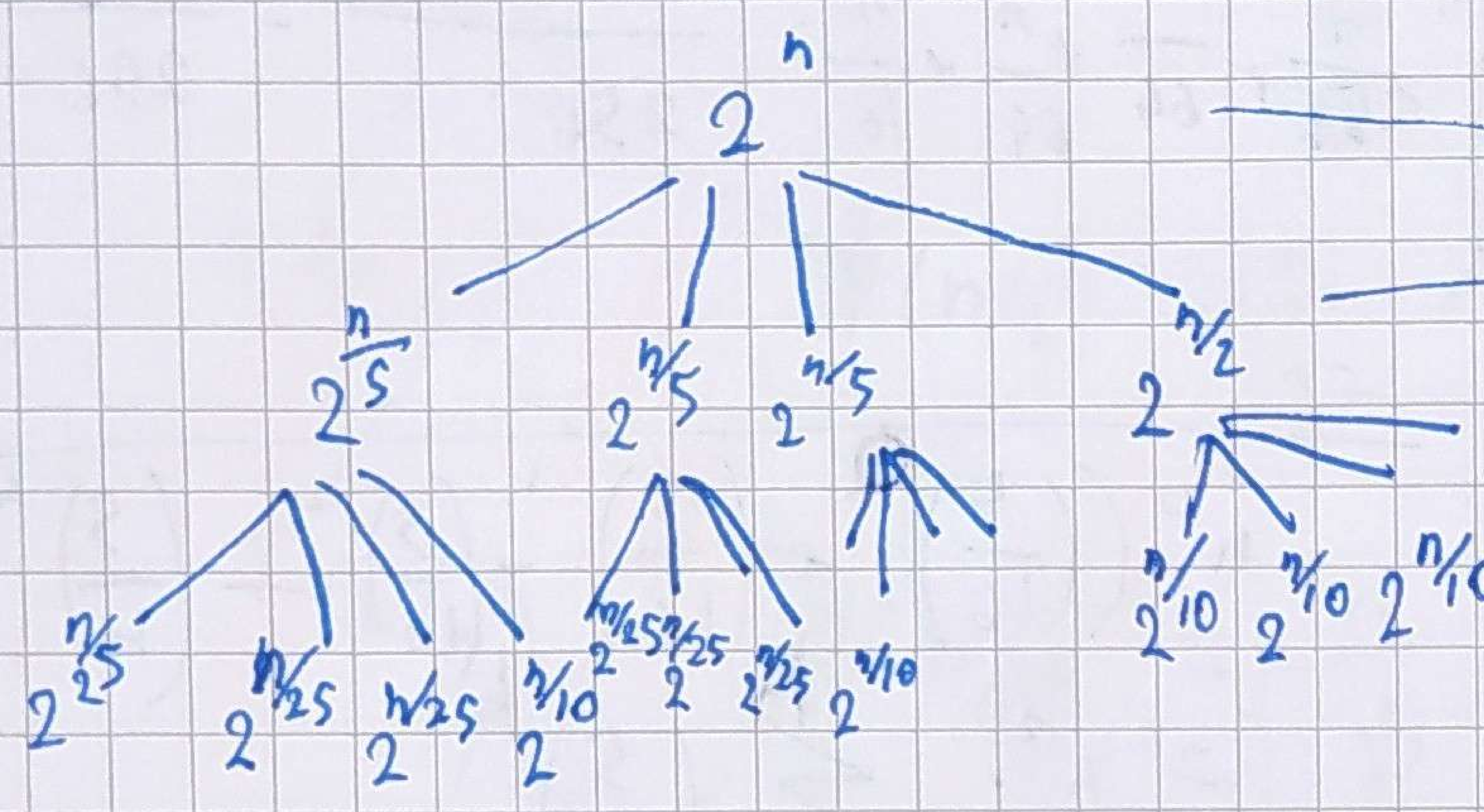
c) $T(n) = 12T(n/2) + n^2 \lg n$ (using master theorem)

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

$n^{\log_b a} = n^{\log_2 12} = n^{3,58} > n^2$

$$\Rightarrow \boxed{T(n) = \Theta(n^{\log_2 12})}$$

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



$\longrightarrow 2^n$

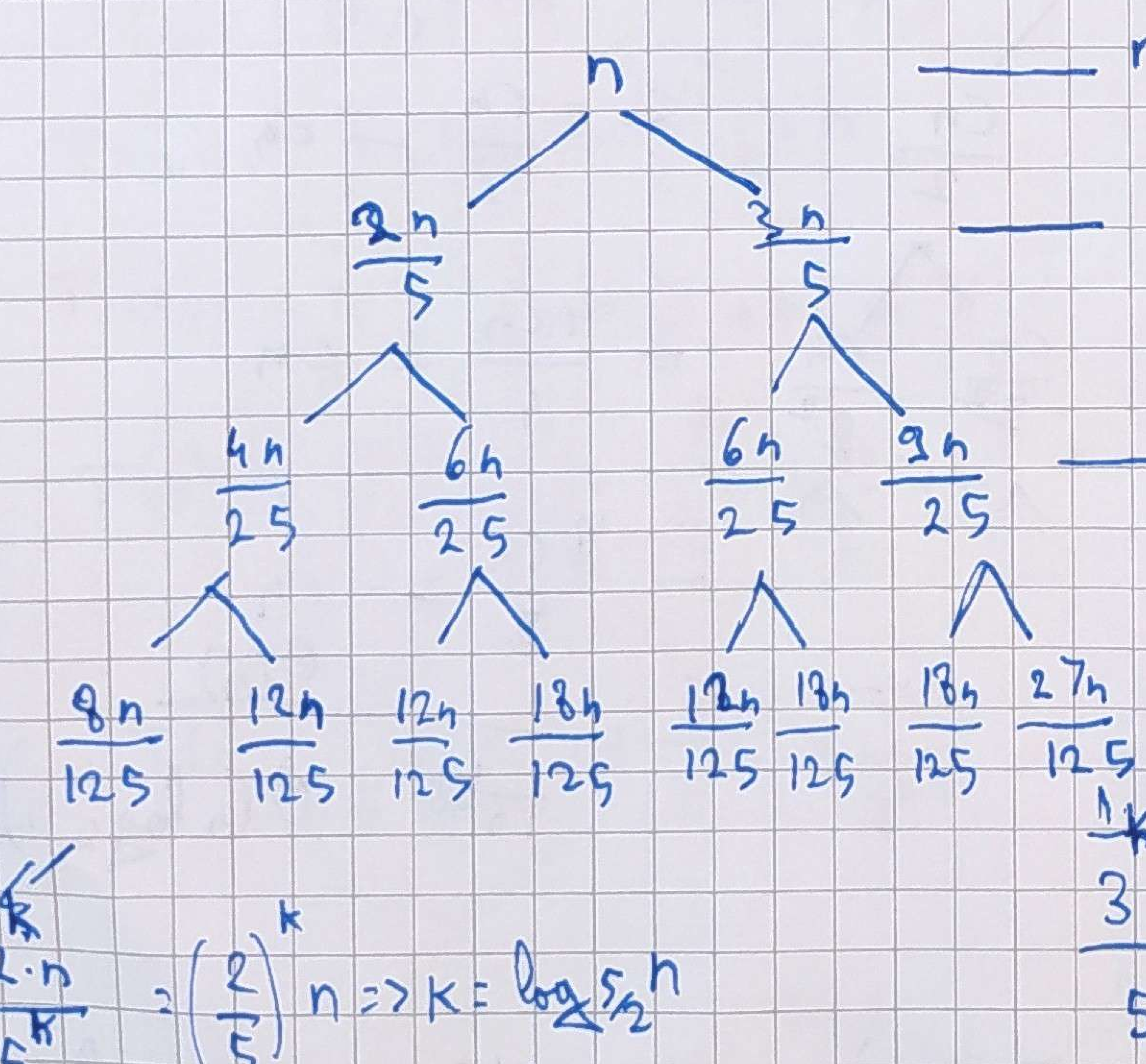
$\longrightarrow 3 \cdot 2^{n/5} + 2^{n/2}$

$\longrightarrow 9 \cdot 2^{n/25} + 6 \cdot 2^{n/10} + 1 \cdot 2^{n/4}$

$$2^n \underbrace{\left(3 \cdot 2^{1/5} + 2^{1/2} + 9 \cdot 2^{1/25} + 6 \cdot 2^{1/10} + 1 \cdot 2^{1/4}\right)}_{C,\ \text{a constant dependent on } n}$$

$$\Rightarrow \boxed{T(n) = C \cdot 2^n \in \Theta(2^n)}$$

e) $T(n) = T(2n/5) + T(3n/5) + \Theta(n)$ (recursion tree)



— n

— $\frac{5n}{5}$

— $\frac{25n}{25}$

— $\frac{125n}{125}$

$h = n \log_5 n$

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

$\frac{3^k n}{5^k} = \left(\frac{3}{5}\right)^k \cdot n \Rightarrow k = \log_{5/3} n$

Best case: $T(n) \in \Omega(n \log_5 n)$

Worst case: $T(n) = \Theta(n \log_{5/3} n)$