Algorithms and Data Structures
Homework 2

Dragi Kamov

Problem 1

a) My algorithm can be found in "merge.cpp".

b) My plot can be found in "plot.xlsx"

c) The worst case and the random case have enlarging times as the k increases, and the best case maintains a low timing which is less than 1s.
The complexity of both the random and worst case will be:

$$\text{Total complexity} = \Theta(k^2 + n\log n)$$

And the best case will have a complexity:

$$\text{Total complexity}: \Theta = (k + n\log n)$$

d) Insertion sort is fast on small arrays, but as the size of the array increases the insertion sort would get slower. Merge sort is faster than insertion in that view so we would have to make a combination between them and keep the k low so that they will have a fast computation.

Problem 2

a) $T(n) = 36T(n/6) + 2n$
I will use the master's method for this reccurrence.

$a = 36$
$b = 6$
$f(n) = 2n$

$n^{\log_6 a} = n^{\log_6 36} = n^2$

If I take $\varepsilon = 0.2$ then I will have:

$$n^{2-0.2} = n^{1.8} \qquad \lim_{n\to\infty} \frac{2n}{n^{1.8}} = \lim_{n\to\infty} \frac{2}{n^{0.8}} = 0$$

From this, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, that means that $f(n)$ is asymptotically smaller than $g(n)$.

$$f(n) = O(g(n)) = O(n^{2-\varepsilon}) = 2n$$

I proved that $f(n) = O(n^{\log_b a - \varepsilon})$ so that means that

$$\underline{\underline{T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)}}$$

b) $T(n) = 5T(n/3) + 17n^{1.2}$

I will use the master's method again.

$a = 5$

$b = 3$

$f(n) = 17n^{1.2}$

$$n^{\log_b a} = n^{\log_3 5} = n^{1.46}$$

If I take $\varepsilon = 0.2$

$$n^{1.46-\varepsilon} = n^{1.46-0.2} = n^{1.26}$$

$$\lim_{n\to\infty} \frac{17n^{1.2}}{n^{1.26}} = \lim_{n\to\infty} \frac{17}{n^{0.06}} = 0$$

So $f(n)$ is asymptotically smaller than $g(n)$

$$f(n) = O(n^{1.46-\varepsilon})$$

$$f(n) = 17n^{1.2} = O(n^{1.46-\varepsilon})$$

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

$$\Rightarrow \underline{\underline{T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_3 5})}}$$

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

Master's method:

$a = 12$

$b = 2$

$f(n) = n^2 \log n$

$$n^{\log_b a} = n^{\log_2 12} = n^{3.58}$$

If I take $\varepsilon = 0.2$

$$n^{3.58-\varepsilon} = n^{3.58-0.2} = n^{3.38}$$

$$\lim_{n\to\infty} \frac{n^2 \log n}{n^{3.38}} = \lim_{n\to\infty} \frac{\log n}{n^{1.38}} = 0$$

So $f(n)$ is asymptotically smaller than $g(n)$

$f(n) = O(n^{3.58-\epsilon}) = n^2 \log n$

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

$\Rightarrow \underline{T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_2 12})}$

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

Substitution method:

We assume that $T(k) \le ck$, where c is a constant.

$T(n/5) < T(n/2)$

$3T(n/5) + T(n/2) < 4T(n/2)$

$4T(n/2) + 2^n \le 4c\frac{n}{2} + 2^n$

$4T(n/2) + 2^n \le 2cn + 2^n$

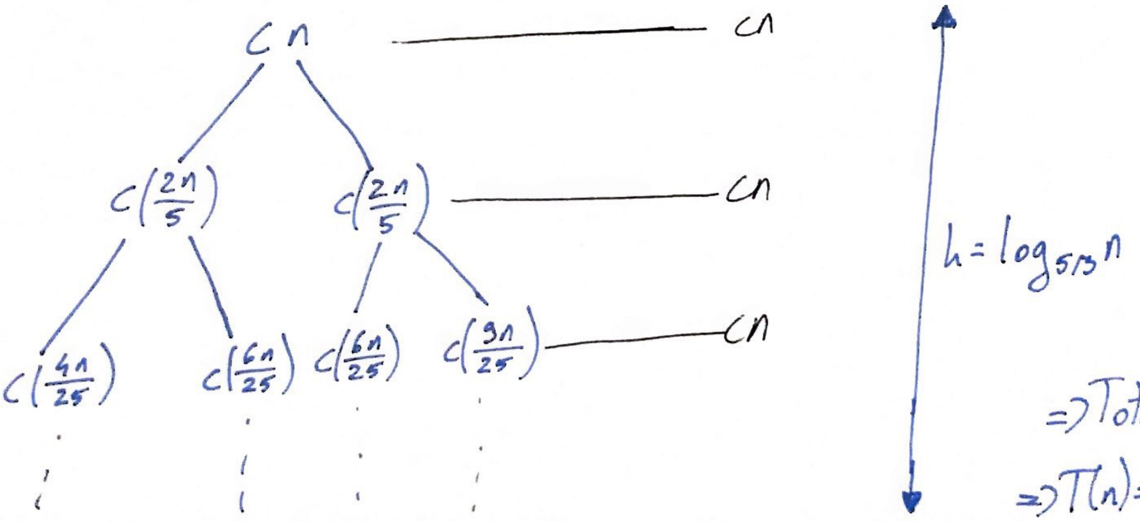
we can say that $T(n) \le 2cn + 2^n$

Time complexity of $2cn + 2^n$ is $\Theta(2^n)$ which means that $T(n) = \Theta(2^n)$

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

Recursion tree:

$f(n) = \Theta(n) \Rightarrow f(n) = cn$



$\Rightarrow \text{Total} = cn\log_{5/3} n$

$\Rightarrow T(n) = O(n \log n)$