Joyan Shandro

Exercise 1.

c) Since we know that $n \lg n$ is asymptotically smaller than n^2, we expect that as the size grows bigger, Merge Sort beats insertion sort.

Since we expect Insertion sort to be faster for small values of array sizes, I have checked ≠ values of k.

From the plots, but also from the times we get when running our program, the average and worst case scenarios take the least (time) for for k=15 and more for ≠k-s.

we also notice that as k increases the computation time increases too. But, for the best case scenarios we notice that in fact the computational time get's better (becomes less) for larger values of k. This happens because in best case the insertion sort, as we proved in class has O(n) complexity. Not only the expectations from the asymptotic analysis, but also the plot proves this.

d) In practice k would be best chosen as 15 as explained above except in those cases in which I expect the array to be almost or already sorted; in these cases k it's better to be taken as the length of the array.

Jovan Shandro

Exercise 2.

a) $T(n) = 36T(n/6) + 2n$. we will apply the master method

$a = 36$, $b = 6$. and $f(n) = 2n$; $n^{\log_b a} = n^2$

and since $f(n) = 2n = O(n^{2-\varepsilon})$ for $\varepsilon = 1 > 0$

then by the master method $T(n) = \Theta(n^2)$

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

$a = 5$, $b = 3$ $\quad n^{\log_3 5} \approx n^{1.46} \geq n^{1.2}$

so $f(n) = 17 \cdot n^{1.2} = O(n^{1.46-\varepsilon})$ for $\varepsilon = 0.2 > 0$

by master method $T(n) = \Theta(n^{\log_3 5})$

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

$a = 12$, $b = 2$ $\quad n^{\log_2 12} \cong n^{3.6} > n^2 \lg n$

and since $n^{1.6}$ is polynomially $\geq \lg n$

we have $f(n) = n^2 \lg n = O(n^{\log_2 12 - \varepsilon})$ for $\varepsilon = 0.5 > 0$

so by master method

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

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

To find an upper bound we will use the fact that

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

so we first solve $T(n) = 4T(n/2) + 2^n$.

by the Master method, since $2^n = \Omega(n^{\log_2 4 + \varepsilon})$ for any $\varepsilon > 0$

since 2^n is exponential and since for $\forall n \geq 2$ we can find $c < 1$ s.t. $4 \cdot 2^{n/2} \leq c \cdot 2^n$ we have $T(n) = \Theta(2^n)$

So we have an upper bound of $c \cdot 2^n$ for some constant c

to find a lower bound we use $T(n) \geq 4T(n/5) + 2^n$

so we solve $T(n) = 4T(n/5) + 2^n$ which by the master method give.

$2^n = \Omega(n^{\log_5 4 + \varepsilon})$ $\forall \varepsilon > 0$ since 2^n is exponential

~~this~~ and since $\forall n > 2$ $\exists c < 1$ s.t. $4 \cdot 2^{n/5} \leq c \cdot 2^n$

we have $T(n) = \Theta(2^n)$

so our $T(n) = O(2^n) = \Omega(2^n) \Rightarrow \boxed{T(n) = \Theta(2^n)}$

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

we will show $T(n) = \Theta(n \lg n)$.

we will show that $\exists\ a, b, c$ constants s. t.

$$a\left(\frac{2n}{5}\right)\lg\left(\frac{n}{5/2}\right) + a\left(\frac{3n}{5}\right)\lg\left(\frac{n}{5/3}\right) + cn \le T(n) \le b\left(\frac{2n}{5}\right)\lg\left(\frac{n}{5/2}\right) + b\left(\frac{3n}{5}\right)\lg\left(\frac{n}{5/3}\right) + cn$$

$$\Updownarrow$$

$$an\lg n - an\left(\frac{2}{5}\lg\frac{5}{2} + \frac{3}{5}\lg\frac{5}{3}\right) + cn \le T(n) \le bn\lg n - bn\left(\frac{2}{5}\lg\frac{5}{2} + \frac{3}{5}\lg\frac{5}{3}\right) + cn$$

since $\frac{2}{5}\lg\frac{5}{2} + \frac{3}{5}\lg\frac{5}{3}$

$= \frac{1}{5}\left(\lg\frac{25}{4} \cdot \frac{25}{9}\right) = \frac{2}{5}\lg\frac{25}{6}$.

in order to have

$$an\lg n \le T(n) \le b\, n\lg n$$

it suffices that a, b, c have the relations

$$c - b \cdot \frac{2}{5}\lg\frac{25}{6} \le 0 \Rightarrow b \ge c \cdot \frac{5}{2} \cdot \frac{1}{\lg\frac{25}{6}}$$

and $$c - a\,\frac{2}{5}\lg\frac{25}{6} \ge 0 \Rightarrow a \le c \cdot \frac{5}{2} \cdot \frac{1}{\lg\frac{25}{6}}$$

So taking 3 constants that have these relations

we have $T(n) = O(n\lg n) = \Omega(n\lg n)$

$\Rightarrow \boxed{T(n) = \Theta(n\lg n)}$