Amine Chahi

Homework 04

Problem 4.1:

a-) check "Main.cpp"

b-) check "Main.cpp"

C-) The Asymptotic time complexity:

for Best Case: time complexity is $O(n)$ in-insertion sort, for different values of K it acquires minimum time, when $k \to +\infty$, only insertion sort is applied

* Average Case: for insertion sort $O(n^2)$ and merge sort is $O(n \log n)$. Hence, we have a complexity of $O\left(\frac{n}{K} \log \frac{n}{K} + K^2\right)$ for all values of K.

* Worst Case: for insertion sort we have a time Complexity $O(n^2)$, merge sort $O(n \log n)$.

Algorithm at first has an asymptotic time complexity of $O\left(\frac{n}{K} \log \frac{n}{K} + K^2\right)$.

As K gets larger complexity becomes $O(n^2)$. (Only insertion sort is applied)

Problem 4.2:

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

Using Master Theorem: we have:

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

Hence $f(n) = O(n^{\log_b a - \epsilon})$ for $\epsilon > 0$

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

By Case 1 of Master Theorem $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

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

We have $a = 5$; $b = 3$; $f(n) = 17n^{1.2}$

$f(n) = \Omega(n^{\log_b a - \epsilon})$ for $\epsilon > 0$ where $\log_3 5 < 1.2$

By Case 3 of Master Theorem

$T(n) = \Theta(f(n)) = \Theta(n^{1.2})$

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

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

we have $n^{\log_b a} = n^{\log_2 12}$

$= n^{3.5a}$

$n^{3.5a} > n^2 \lg n \Rightarrow$ ~~[illegible]~~ It will polynomially increase for some value and ϵ can be found

By Case III of Master theorem $T(n) = \Theta(f(n))$

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

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

Using recursion tree method

At each level we have sum of $T(n/5)$ and $T(n/2)$

Hence, Cost at each level is $2n$. Number of levels is $\log_{5/2} n$, ~~since~~ as problem size decreases by a factor of $5/2$ at each iteration

$\Longrightarrow T(n) = \Theta(n \log_{5/2} n)$.

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

Using recursion Tree method:

At each level, we have two recursive calls: for $2n/5$ and $3n/5$. The cost is $\Theta(n)$ at each level due to the constant term.

We Can Observe that the cost increases linearly with n, while number of levels grows logarithmically with n ~~due to the problem~~ due to problem size decreasing by a constant fraction

* Hence, ~~we can conclude~~ since the number of levels grows logarithmically, Total Cost is $O(n \log n)$

* For lower Bound, we observe number of leaves in the recursion tree is at least $n^{\log_{5/3} 2}$, ~~giving a lower bound~~

giving a Lower Bound of $\Omega(n \log n)$

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