Mahiem Agrawal
Assignment 4

Problem 4.1

a) Implementation inside the file "Mergesort.cpp" according to the question.

b) Graph implemted for various values of k from 0 till the size.

c) ~~The~~ Best-case.

For the different values of k, we can see that the best case requires less time. This is because the time complexity of best case in insertion sort is just $\Theta(n)$ and when k becomes large only insertion sort is applied hence the complexity $\Theta(n)$. but for

Average-case.

The average case for merge sort is $\Theta(n \log n)$ and insertion sort is $\Theta(n^2)$ therefore our algorithms complexity is $\Theta(n \log n/k + k^2/k)$ and this is true for all various values of k.

c) worst-case.

The worst case for insertion sort is $\Theta(n^2)$ but merge sort is just $\Theta(n \log n)$. Initially they will have complexity of $\Theta(n/k \log n/k + k^2)$ however as k gets lower just insertion sort is applied so the complexity becomes $\Theta(n^2)$.

d) This would depend on the type of array I have. If the array is already corted I would have a large value of k so only insertion sort is applied.

If the array is unsorted, I would chose the value of k to be 1 as this way just the complexity would be just $\Theta(n \log n)$.

Problem 4.2.

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

Using master method.

$T(N) = aT(n/b) + f(n)$

$a = 36$

$b = 6$

$f(n) = 2n$.

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

$\therefore f(n) = O(n^{2-E})$ for $E = 1$ so,

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

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

$a = 5$

$b = 3$

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

$n^{\log_b a} = n^{1.46}$.

$f(n) = O(n^{1.46 - E})$ for $E = 0.26$)

$\therefore T(n) = \Theta(n^{1.46})$.

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

$a = 12$

$b = 2$

$f(n) = n^2 \lg n$.

Now,

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

$= n^{3.59}$.

Now,

Now we can use case (I)

as

$n^{3.5a} > n^2 \lg n$ will be polynomially greater for some value and we can easily find some constant G.

$\therefore$

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

$= \Theta(n^{3.5a})$