Question 4,

We know that, cut a list with n elements into n/k sub-lists of length k
a, the insertion sort can sort a list of length k in $\Theta(k^2)$, so for n/k sub-lists, the total time will be $\Theta(k^2*(n/k))= \Theta(nk)$

b, since the depth of recursion tree for normal merge sort is Log(n),
so the depth of recursion tree for this problem is Log(n)- Log(k), because the recursion tree stops at the sub-list at size k. therefore, the depth of recursion tree is log(n/k)
and each level time consuming is cn, so the total time is log(n/k)*cn which is equals to $\Theta(nlog(n/k))$.

c, if nlog(n/k)+nk = n log(n) in terms of Θ notation
then we need to make sure
nlog(n/k) = Θ(nlogn) (if nlog(n/k) >nk) ~1
or
nk= Θ(nlogn) if nlog(n/k) <nk ~2
so for 1,
k is a constant, not a function of n
for 2,
k =Θ(logn), they have the same asymptotic.
So k =Θ(logn), or we can say k $\in O$ (logn)

d, in practice, we can choose k depends on value c1 and c2 which are the coefficients of nk and nlog(n/k) hidden by the asymptotic notation. We can build a mathematic model to analysis it by doing calculus 1 of k, and find the peak value which makes (nk+ nlog(n/k))' = 0, then we can choose the k value, again the values of coefficients are important for choosing k value.