Although merge sort runs in Θ(nlogn) worst-case time and insertion sort runs in Θ(n2) worst-case time, the constant factors in insertion sort can make it faster in practice for small problem sizes on many machines. Thus, it makes sense to **coarsen** the leaves of the recursion by using insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which n/k sublists of length k are sorted using insertion sort and then merged using the standard merge mechanism, where k is a value to be determined.

1. Show that insertion sort can sort the n/k sublists, each of length k, in Θ(nk) worst-case time.
2. Show how to merge the sublists in Θ(nlg(n/k)) worst-case time.
3. Given that the modified algorithm runs in Θ(nk+nlg(n/k)) worst-case time, what is the largest value of k as a function of n for which the modified algorithm has the same running time as standard merge sort, in terms of Θ-notation?