Due at the start of class Thursday, February 13.

When running Merge Sort, it is quite likely advantageous to use Selection Sort (rather than continue the recursive calls of Merge Sort) when the size of the list being sorted gets small enough. Assume we run Merge Sort on a list of size n, but when the list has size $\leq m$ we use Selection Sort. Merge Sort could be considered the special case of this algorithm for $m = 1$.

Problem 1. Write pseudo-code for this generalized Merge Sort algorithm. You can assume that the Merge and Selection Sort routines are given to you (so you do not have to write them).

Problem 2. Write a recurrence for the number of comparisons.

Problem 3. How many comparisons are done by an single invocation of Merge Sort at level i, where the original call is at level 0. (In other words, i is the number of times the word "vice" occurs before the word "president" in the description of the algorithm as done in class.)

Assume that the list size is exactly m times a power of 2 (i.e. $n = 2^k m$, for some integer k). Note that there is a special case for level k.

Problem 4. How many comparisons are done by all invocations of Merge Sort at level i. As above there is a special case for level k.

Problem 5. Write a summation for the total number of comparisons in this generalized Merge Sort; solve the summation; and simplify. Show your work.

Ideally, your final answer should only involve n and m. You should write your final answer as a term involving $n \lg n$, followed by a linear term (in n), followed by a constant term. (You can check that your answer is correct by substituting $m = 1$.)

Problem 6. (Challenge Problem) Assume that there is an additive constant α associated with each (recursive) procedure call, a multiplicative constant β associated with Merge, and a multiplicative constant γ associated with Selection Sort. Calculate the optimal value of m. [This problem will not count towards your grade on this homework.]