

1) (10 pts) ANL (Algorithm Analysis)

Consider the problem of taking n sorted lists of n integers each, and combining those lists into a single sorted list. For ease of analysis, assume that n is a perfect power of 2. Here are two potential algorithms to solve the problem:

Algorithm A: Run the Merge Algorithm, defined between two lists, on lists 1 and 2 to create a single merged list. Then rerun the algorithm on this merged list and list 3, creating a merged list of all items from lists 1, 2 and 3. Continue in this fashion, running the Merge Algorithm $n-1$ times, always between the currently "growing" list and the next list to be merged into it, until list n is merged in, creating a single sorted list.

Algorithm B: Pair up the lists into $\frac{n}{2}$ pairs of lists of size n . Run the Merge Algorithm on each of these pairs. Once this phase finishes, there will be $\frac{n}{2}$ lists with $2n$ integers. With the new lists, repeat the process until we are left with a single sorted list.

With sufficient work and proof, determine the Big-Oh run time, in terms of n , of both of these algorithms. Clearly put a box around your final answer for both algorithms.