Task-3

The function dvd_nd_conqr recursively takes a list of integers data and an optional argument int. It returns a tuple containing two values: a list of sorted integers and the number of inversions in that list. If the length of the input list is 0 or 1, return the list and 0 as the number of inversions. Otherwise, divide the input list into two halves and recursively call into ~~a single~~ dvd_nd_conqr on each half. Merge the two sorted lists returned by each recursive call into a single sorted list, while counting the number of inversions between them. The time complexity of this algorithm can be expressed as $T(n) = 2T(n/2) + O(n)$, where n is the length of the input list. This recurrance relation can be solved using the master theorem, which gives a time complexity of $O(n \log n)$ for this algorithm.