CITS2200 Project

Name: Ciaran Engelbrecht
Student ID: 23169641

Fines:

My implementation of Fines incorporated the Merge Sort algorithm, in order to count the smaller numbers following the current indexed value and incrementing this value in order to calculate the total number of fines which should be returned. This value is calculated when each inversion is performed through the Merge Sorting function. There is a counter variable which records how many values from the right array we have added, with another array keeping a record of this result, and when a value is moved from the right array into the new array this counter is incremented. And when a number from the left array is moved into the new array, the index of the number in the count array is increased by the right count variable. Thus, returning the total number of fines. This is completed with a computational complexity of O(n Log n), since the entire input will be iterated through in worst case with n items iterated log n times. This will work in every case as the code will iterate over each value in the query, counting the total number of values which are lesser than it, following its index and increase the count by as much if the ship failed to give way to any ships at all.

https://stackoverflow.com/questions/337664/counting-inversions-in-an-array
https://algo.monster/problems/count_of_smaller_numbers_after_self