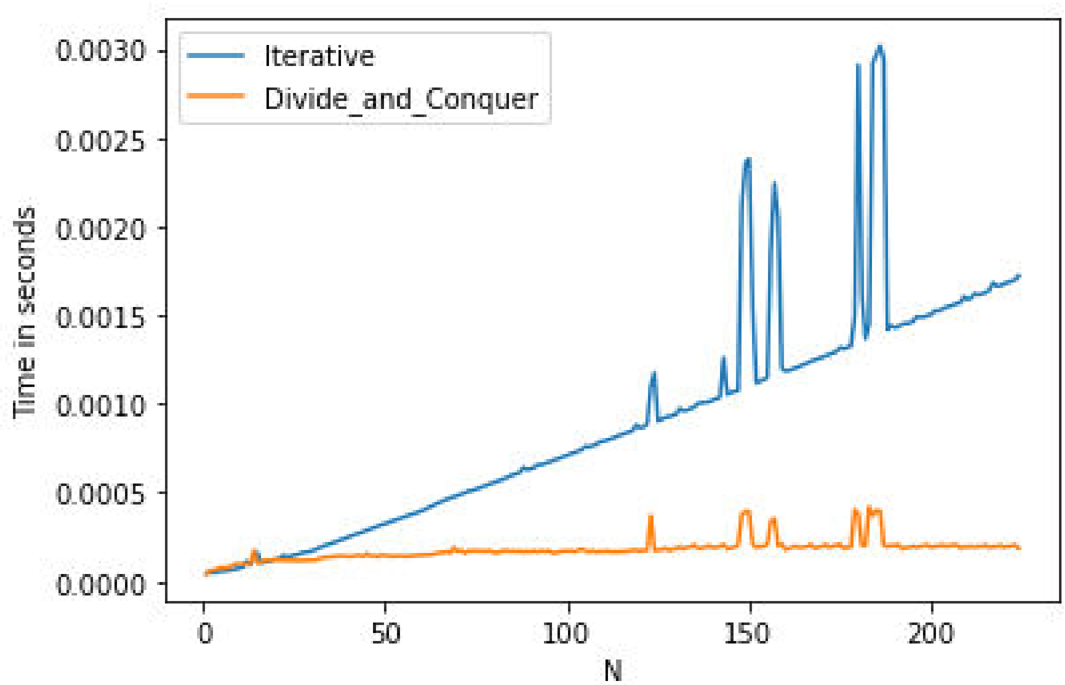


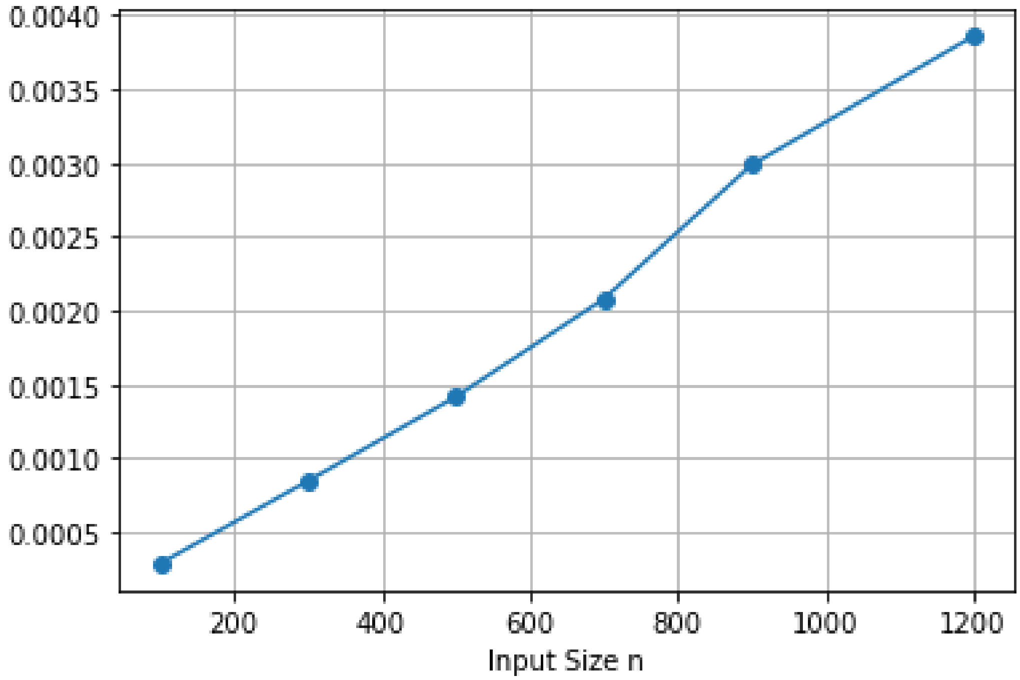

(B)(figure above in Question 1)

1-The iterative runs in O(n) time as it keeps multiplying the base until n times in a one loop(Blue Line)

2-A recurrence relation is used to test the Divide-and-Conquer Method. T(n) denotes the temp complexity of D and C Method,When n is even, T(n) = T(n/2) + O(1); when n is odd, T(n) = T((n-1)/2) + O(1),This recurrence can be solved by using the Master Theorem where a = 1, b = 2, and k = 0. Thus, O(log n) is the time complexity.(Orange Line)

Note:the Laptop couldnt Compute Large Values and Plot Them But What Work With Smaller Set Will Always Work will Larger Ones

Q2:




for Merge sort: algorithm has a time complexity of O(n log n) in the worst case.

for Binary search: the worst case, it has a time complexity of O(log n).

In your algorithm, time complexity is O(n log n) due to binary search is called for each element of the sorted array

merge sort operation is the dominant factorthe resulting in overall time complexity of (O(n log n))

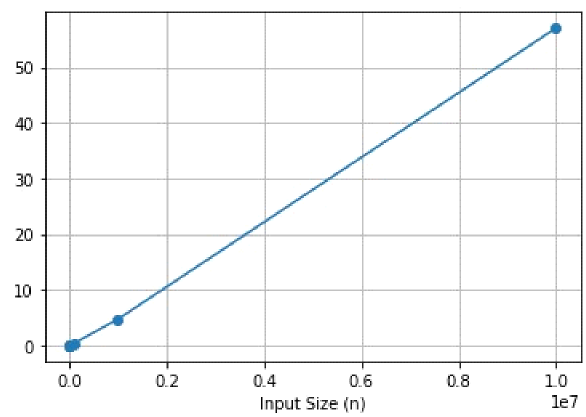
Therefore, the time complexity of merge sort is Θ(n * log(n)).

The binary search component, which is O(log n), does not change the overall complexity of the merge sort.

Note: I Used Smaller Data Set thats the maximum My Laptop Can Handle What Work in Small Sets Work in Bigger Ones

Note (the only Plot Proccessed By My Laptop )

Q2:


50
40
30
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0
1e7
Input Size (n)