Merge Sort: In merge sort we divide the master array into two subarrays and merging the subarrays. Time taken to merge sort subarrays will be T(n/2) *2 if we assume time taken to merge sort master array as T(n) and time taken to merge will be cn. Hence time taken is given by T(n)=T(n/2) +cn. Time complexity is n* log_2 (n)

Insertion Sort: There can be a maximum of n*(n-1)/2 inversions while implementing insertion sort. Hence the time complexity of insertion sort is O(n^2).

Selection sort: In selection sort we implement nested loops. So, the total number of loops implemented will be summation of n (n*(n+1)/2). Hence the time complexity is O(n^2).

Bubble sort: similar to selection sort we implement nested loops here also but total number of loops implement will be summation of n-1 (n*(n-1)/2). Hence the time complexity is O(n^2).

Binary Search: Binary Search has a similar algorithm to Merge sort and hence the time complexity is n*log_2(n)

Linear search: In linear search the elements are compared one by one to check if the element is present. Hence the time taken will be cn where c is the time taken to check 1 index . Hence the time complexity is O(n).