**Which sorting algorithm will take least time when all elements of input array are identical? Consider typical implementations of sorting algorithms.**

Insertion Sort

**Which of the following is not a stable sorting algorithm in its typical implementation.**

Quick Sort

**Consider a situation where swap operation is very costly. Which of the following sorting algorithms should be preferred so that the number of swap operations are minimized in general?**

Selection Sort

**Which of the following is not true about comparison based sorting algorithms?**

TODOS SON VERDADEROS  
The minimum possible time complexity of a comparison based sorting algorithm is O(nLogn) for a random input array  
Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared

**What is the best time complexity of bubble sort?**

N

**What is the worst case time complexity of insertion sort where position of the data to be inserted is calculated using binary search?**

N^2

**The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of**

NlogN

**In a modified merge sort, the input array is splitted at a position one-third of the length(N) of the array. Which of the following is the tightest upper bound on time complexity of this modified Merge Sort.**

 N(logN base 2/3)

**A list of n string, each of length n, is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is**

O (n2 log n)