Quiz on Sorting Anwers With Explanation wherever possible.

Q1 Assume that a merge sort algorithm in the worst case takes 30 seconds for an input of size 64.Which of the following most closely approximates the maximum input size of a problem that can be solved in 6 minutes?

1) 256
2) 512
3) 1024
4) 2048

Answer is option 2 ie 512

Explanation: Time complexity of merge sort is Θ(nLogn)

c*64Log64 is 30 where c is constant.

c*64*6 is 30

c is 5/64

For time 6 minutes

5/64*nLogn = 6*60

nLogn = 72*64 = 512 * 9

n = 512.

2) Which of the following changes to typical QuickSort improves its performance on average and are generally done in practice?
1) Randomly picking up to make worst case less likely to occur.
2) Calling insertion sort for small sized arrays to reduce recursive calls.
3) QuickSort is tail recursive, so tail call optimizations can be done.
4) A linear time median searching algorithm is used to pick the median, so that the worst case time reduces to O(n Log n)

1) 1 and 2
2) 2, 3 and 4
3) 1, 2 and 3
4) 2, 3 and 4

Answer is option3) ie 1,2 and 3

Q3) Randomized quicksort is an extension of quicksort where the pivot is chosen randomly. What is the worst case complexity of sorting n numbers using randomized quicksort?

O(n^2)

O(n)

O(nlogn)

Answer: O(n^2)
Explanation: Randomized quicksort has expected time complexity as O(nLogn), but worst case time complexity remains same. In worst case the randomized function can pick the index of corner element every time.

Q4) In a permutation a1…..an of n distinct integers, an inversion is a pair (ai, aj) such that i < j and ai > aj. What would be the worst case time complexity of the Insertion Sort algorithm, if the inputs are restricted to permutations of 1…..n with at most n inversions?

O(n^2)

O(nlogn)

O(n^1.5)

O(n)

Answer: O(n)

Explanation: Insertion sort runs in Θ(n + f(n)) time, where f(n) denotes the number of inversion initially present in the array being sorted.

Q5) Let P be a QuickSort Program to sort numbers in ascending order using the first element as pivot. Let t1 and t2 be the number of comparisons made by P for the inputs {1, 2, 3, 4, 5} and {4, 1, 5, 3, 2} respectively. Which one of the following holds?
(A) t1 = 5
(B) t1 < t2
(C) t1 > t2
(D) t1 = t2

Answer: (C)

Explanation: When first element or last element is chosen as pivot, Quick Sort's worst case occurs for the sorted arrays.

Q6 Which of the following is an external sorting?

a) Insertion Sort
b) Merge Sort
c) Bubble Sort
d) Tree Sort

Answer b) Merge Sort

Explanation: Merge sort algorithm, sorts chunks that each fit in RAM, then merges the sorted chunks together. We first divide the file into runs such that the size of a run is small enough to fit into main memory. Then sort each run in main memory using merge sort sorting algorithm. Finally merge the resulting runs together into successively bigger runs, until the file is sorted.

Q7 Partition and exchange sort is ...

a) QuickSort
b) Heap Sort
c) Tree Sort

Answer is a) QuickSort

Q8 The usual O(n^2) implementation of insertion sort uses the linear search to identify the position where an element is to be inserted into the already sorted part of the array. If instead, we use binary search to identify the position the worst case running time will be:

1) O(n)
2) O(nlogn)
3) O(n^2)
4) O(n(logn)^2)

Answer is 3

Q9 If quicksort is written so that partition algorithm always uses the median value of the segment as the pivot then worst case performance of quick sort will be?

1) O(n)
2) O(nlogn)
3) O(n^2)

Answer is 2) O(nlogn)

Explanation: Median value is the middle value in sorted Array.

Q10 You have an array of n elements. Suppose you implement quicksort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is

1) O(n^2)
2) O(nlogn)
3) O(n)

Answer is 1) O(n^2)

Explanation: Whichever element we take as Pivot, either first or middle, worst case will be $O(n^2)$ since Pivot is fixed in position.