GeeksQuiz

Computer science mock tests for geeks

Sorting

Question 1

What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?

A Recurrence is T(n) = T(n-2) + O(n) and time complexity is O(n^2)

B Recurrence is T(n) = T(n-1) + O(n) and time complexity is O(n^2)

C Recurrence is T(n) = 2T(n/2) + O(n) and time complexity is O(nLogn)

D Recurrence is T(n) = T(n/10) + T(9n/10) + O(n) and time complexity is O(nLogn)

Discuss it

Question 2

Suppose we have a O(n) time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort.

A O(n^2 Logn)

B O(n^2)

C O(n Logn Logn)

D O(nLogn)

Discuss it

Question 3

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

A Insertion Sort

B Merge Sort

C Quick Sort

D Bubble Sort

Discuss it

Question 4

Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced).

A Quick Sort

B Heap Sort

C Merge Sort

D Insertion Sort

Discuss it

Question 5

Given an unsorted array. The array has this property that every element in array is at most k distance from its position in sorted array where k is a positive integer smaller than size of array. Which sorting algorithm can be easily modified for sorting this array and what is the obtainable time complexity?

A Insertion Sort with time complexity O(kn)

B Heap Sort with time complexity O(nLogk)

C Quick Sort with time complexity O(kLogk)

D Merge Sort with time complexity O(kLogk)

Discuss it

Question 6

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?

A Heap Sort

B Selection Sort

C Insertion Sort

D Merge Sort

Discuss it

Question 7

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

A The minimum possible time complexity of a comparison based sorting algorithm is O(nLogn) for a random input array

B Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared

C Counting Sort is not a comparison based sorting algortihm

D Heap Sort is not a comparison based sorting algorithm.

Discuss it

Question 8

Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 2 5 1 7 9 12 11 10 Which statement is correct?

A The pivot could be either the 7 or the 9.

B The pivot could be the 7, but it is not the 9

C The pivot is not the 7, but it could be the 9

D Neither the 7 nor the 9 is the pivot.

Discuss it

Question 9

Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this: 16 14 15 10 12 27 28 How many heapify operations have been performed on root of heap?

A 1

B 2

C 3 or 4

D 5 or 6

Discuss it

Question 10

What is the best time complexity of bubble sort?

A N^2

B NlogN

C N

D N(logN)^2

Discuss it

Question 11

You have to sort 1 GB of data with only 100 MB of available main memory. Which sorting technique will be most appropriate?

A Heap sort

B Merge sort

C Quick sort

D Insertion sort

Discuss it

Question 12

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

A N

B NlogN

C N^2

D N(logN)^2

Discuss it

Question 13

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

sorting is of the order of

A N

B N^2

C NlogN

D N(logN)^2

Discuss it

Question 14

In a modified merge sort, the input array is splitted at a position one-third of the length(N) of the array. What is the worst case time complexity of this merge sort?

A N(logN base 3)

B N(logN base 2/3)

C N(logN base 1/3)

D N(logN base 3/2)

Discuss it

Question 15

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

A Insertion Sort

B Heap Sort

C Merge Sort

D Selection Sort

Discuss it

Question 16

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 (A) $O(nlogn$ (B) $O(n^2logn)$ (C) $O(n^2 + logn)$ (D) $O(n^2)$

A A

B B

C C

D D

Discuss it

Question 17

In quick sort, for sorting n elements, the (n/4)th smallest element is selected as pivot using an O(n) time algorithm. What is the worst case time complexity of the quick sort? (A) θ(n) (B) θ(nLogn) (C) θ(n^2) (D) θ(n^2 log n)

A A

B B

C C

D D

Discuss it

Question 18

Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then

A T(n) <= 2T(n/5) + n

B T(n) <= T(n/5) + T(4n/5) + n

C T(n) <= 2T(4n/5) + n

D T(n) <= 2T(n/2) + n

Discuss it

Question 19

Which of the following sorting algorithms has the lowest worst-case complexity?

A Merge Sort

B Bubble Sort

C Quick Sort

D Selection Sort

Discuss it

Question 20

Which sorting algorithms is most efficient to sort string consisting of ASCII characters?

A Quick sort

B Heap sort

C Merge sort

D Counting sort

Discuss it

Question 21

The number of elements that can be sorted in $\Theta(logn)$ time using heap sort is

(A) $\Theta(1)$
(B) $\Theta(\sqrt{logn})$
(C) $\Theta(Logn/(LogLogn))$
(d) $\Theta(Logn)$

A A

B B

C C

D D

Discuss it

Question 22

Which of the following is true about merge sort?

A Merge Sort works better than quick sort if data is accessed from slow sequential memory.

B Merge Sort is stable sort by nature

C Merge sort outperforms heap sort in most of the practical situations.

D All of the above.

Discuss it

Question 23

Given an array where numbers are in range from 1 to n^6, which sorting algorithm can be used to sort these number in linear time?

A Not possible to sort in linear time

B Radix Sort

C Counting Sort

D Quick Sort

Discuss it

Question 24

In quick sort, for sorting n elements, the (n/4)th smallest element is selected as pivot using an O(n) time algorithm. What is the worst case time complexity of the quick sort? <pre> (A) θ(n) (B) θ(nLogn) (C) θ(n^2) (D) θ(n^2 log n) </pre>

A A

B B

C C

D D

Discuss it

Question 25

Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then

A T(n) <= 2T(n/5) + n

B T(n) <= T(n/5) + T(4n/5) + n

C T(n) <= 2T(4n/5) + n

D T(n) <= 2T(n/2) + n

Discuss it

Question 26

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

Discuss it

Question 27

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

A $O(n^2)$

B O(nLogn)

C Theta(nLogn)

D $O(n^3)$

Discuss it

Question 28

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?

A $\Theta(n^2)$

B $\Theta(n \log n)$

C $\Theta(n^{1.5})$

D $\Theta(n)$

Discuss it

Question 29

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?

A O(n)

B O(n Log n)

C $O(n^2)$

D O(n!)

Discuss it

Question 30

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(nLogn)
```

A 1 and 2

B 2, 3, and 4

C 1, 2 and 3

D 2, 3 and 4

Discuss it

Question 31

Which one of the following is the recurrence equation for the worst case time complexity of the Quicksort algorithm for sorting n(≥ 2) numbers? In the recurrence equations given in the options below, c is a constant.

A $T(n) = 2T(n/2) + cn$

B $T(n) = T(n - 1) + T(0) + cn$

C $T(n) = 2T(n - 2) + cn$

D T(n) = T(n/2) + cn

Discuss it

Question 32

Assume that a mergesort 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?

A 256

B 512

C 1024

D 2048

Discuss it

There are 32 questions to complete.