6 Searching and Sorting

1. The average number of key comparisons done in a unsuccessful sequential search in a list of length n is:
   (A) $\log n$
   (B) $\frac{n-1}{2}$
   (C) n
   (D) $\frac{n}{2}$
   (E) $\frac{n+1}{2}$

2. Which of the following recurrence relations represents Linear search scheme?
   (A) $T(n) = T(n-1) + T(n-2) + O(1)$
   (B) $T(n) = 2 * T(n-1) + O(1)$
   (C) $T(n) = T(n-1) + O(1)$
   (D) $T(n) = T(n-1) + O(n)$
   (E) $T(n) = T(\frac{n}{2}) + O(1)$

3. In which of the cases shown below, *Binary search* can not be applied for searching?
   (A) Hierarchical data record
   (B) Internet Domain name conversion
   (C) Searching a telephone number in directory.
   (D) A random array of integers

4. What is the best and the worst case complexities of linear search algorithm
   (A) O(nlogn), O(nlogn)
   (B) O(logn), O(nlogn)
   (C) O(n), O(n)
   (D) O(1), O(n)
   (E) O(1), O(1)

5. Which of the following is a not an advantage of linear search algorithm?
   (A) Smaller time complexities compared to binary search
   (B) Greater time complexities compared to binary search
   (C) Difficult to understand
   (D) Wider range of application than binary search
   (E) All of the above

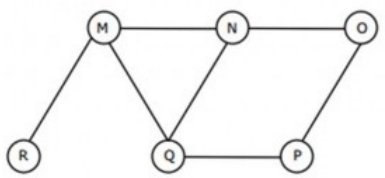
6. Consider following piece of code which tries to find out all duplicate elements in an array of integers with one line missing as follows:
   arr_length is defined as length of the given array.

```
int i;
for (i = 0; i <arr_length-1; i++){
for (int j = i+1; j <arr_length; j++){
if( (———){
printf("%d",arr[i]);
}
}
}
```

What could be a better replacement for the blank line shown above?
(A) arr[i]==arr[j]) ||| (i != j)
(B) arr[i]!=arr[j]) ||| (i == j)
(C) arr[i]==arr[j]) && (i != j)
(D) arr[i]!=arr[j]) && (i != j)

7. Which of the following recurrence relations represents the worst case behavior of binary search?
(A) $T(n) = T(\frac{n}{2}) + O(n)$
(B) $T(n) = 2 * T(n-1) + O(1)$
(C) $T(n) = T(n-1) + O(1)$
(D) $T(n) = 2 * T(\frac{n}{2}) + O(1)$
(E) $T(n) = T(\frac{n}{2}) + O(1)$

8. The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



(A) MNOPQR
(B) NQMPOR
(C) QMNPRO
(D) QMNPOR

9. Given an array of integers what is the minimum number of comparisons needed to find the second largest element in the array ?
(A) $n + 1$
(B) $n + logn + 2$
(C) $2n - 3$

(D) $n + logn - 2$

10. Consider a sorted array of n integers. We are to find a pair of two integers in the array which differ by d, i.e. $|a - b| = d$. The complexity of the best algorithm which solves this problem is:
(A) O(n)
(B) O(nlog n)
(C) O(n^2)
(D) O(logn)

11. Let A be an array of 63 elements consisting of a sequence of 0's followed by a sequence of 1's followed by a sequence of 2's. If the smallest and the largest indices are i and j, such that $A[i] = 1 \& A[j] = 1$ then in the worst case number of comparisons performed by an optimal algorithm is:
(A) 5
(B) 7
(C) 9
(D) 11
(E) 13

12. What is a right interpretation of the term *in-place sorting algorithm*?
(A) It needs $O(1)$ memory to create auxiliary locations
(B) It requires $O(n)$ memory to create auxiliary locations
(B) The array is sorted without using any memory at all
(C) It requires additional storage
(D) None of the mentioned

13. What is a right interpretation of the term *external sorting algorithm*?
(A) Algorithm that uses memory while executing
(B) Algorithm that uses tape or disk while executing
(C) Algorithm that does not need any additional memory to operate
(D) Algorithm that are not *in place*.

14. What is a right interpretation of the term *stable sorting algorithm*?
(A) Algorithm that uses constant amount of memory while executing
(B) Algorithm that has completed its execution
(C) Algorithm that does not change the relative position of duplicates
(D) Algorithm that does not allow duplicates in input array.

15. What is the best case time complexities of selection, insertion and quick sort in that order?
(A) $O(nlogn), O(nlogn), O(nlogn)$

(B) $O(n^2), O(nlogn), O(nlogn)$
(C) $O(n^2), O(n), O(nlogn)$
(D) $O(n^2), O(n^2), O(n^2)$

16. Which of the following statements about comparison based sorting algorithms is correct:
(A) Insertion sort is better to use when input array is partially sorted
(B) Selection sort is as bad as Quick sort in worst case
(C) Bubble sort is preferred when the input size is sufficiently large.
(D) Quick sort outperforms Heap sort due to hidden overhead of computing.

17. Consider an improved version of standard Bubble sort algorithm.
In standard Bubble sort algorithm if a Boolean variable, *flag*, is used to detect absence of inversions we can improve the performance of Bubble sort. What is the best case complexity of bubble sort in the improvised version?
(A) O(nlogn)
(B) O(logn)
(C) O(n)
(D) $O(n^2)$

18. In quick sort, for sorting an array with n elements, arithmetic mean of element in array is selected as pivot using some $O(n)$ time algorithm followed by standard quick sort algorithm.
Which of the following is the worst case time complexity of this quick sort algorithm?
(A) $\Theta(n)$
(B) $\Theta(nLogn)$
(C) $\theta(n^2)$
(D) $\theta(n^2logn)$

19. Which of the following sorting algorithm best suited for an array of integers which is at least 50% sorted.
(A) Bubble Sort
(B) Selection Sort
(C) Quick Sort
(D) Insertion Sort
(E) Merge Sort
(F) Heap Sort

20. Which of the following standard sorting algorithms is not stable:
(A) Bubble Sort

(B) Merge Sort
(C) Insertion Sort
(D) Quick Sort

21. In a system where memory is quite small than a program which tries to sort a chunk of data which is big enough to not fit in memory at once, which of the sorting algorithm may be helpful:
(A) Bubble Sort (with Flag variable)
(B) Merge Sort
(C) Randomized Quick Sort
(D) Heap Sort
(E) Selection Sort
(F) Insertion Sort

22. In Insertion sort, instead of using linear search if binary search is used it can enhance the search time of the algorithm effectively.
What is the worst case time complexity of this Insertion sort algorithm at worst case?
(A) n
(B) $nlogn$
(C) n^2
(D) $nlogn^2$

23. Consider a version of Merge sort algorithm:
The given array is divided such that instead of dividing the array in two halves, it gets divided into 2 parts such that one part contains 30% of elements while other part contains 70% of the total elements. Rest of the procedure remains consistent with the standard merger sort.
Which of the following asymptotic notations represents the closest time complexity for this version of merge sort?
(A) $O(n^{0.7}\log_2 n)$
(B) $O(n\log_{\frac{7}{10}} n)$
(C) $O(n\log_2 n)$
(D) $O(n\log_{\frac{10}{7}} n)$

24. Consider an array of integers:
A = 3,9,10,11,15,17,22,52,43,42,60,70. Bubble sort is used to sort this array. How many iterations improvised version of Bubble sort, i.e. Bubble sort with flag variable, will make?
(A) 12
(B) 11
(C) 10
(D) 5

(E) 3

25. In scenarios where *swap()* operation is a deciding factor to select any sorting algorithm which algorithm can be best suited in such a situation?
(A) Insertion Sort
(B) Selection Sort
(C) Bubble sort
(D) Quick Sort
(E) Merge Sort

26. Write the worst case complexities of the following Sorting algorithms:
(I) Bubble Sort
(II) Merge Sort
(III) Quick Sort
(IV) Heap Sort
(V) Selection Sort
(VI) Insertion Sort

27. If in a given array of size n, pivot is selected such that it divides the array into 2 parts such that one part contains $O(\frac{n}{10})$ elements while other part contains $O(\frac{9n}{10})$ elements of the array, which of the following recurrence relations explains the behavior of this Quick sort?
(A) $T(n) = 2T(\frac{n}{2}) + O(n)$
(B) $T(n) = 2 * T(\frac{n}{10}) + T(\frac{9n}{10}) + O(n)$
(C) $T(n) = T(\frac{n}{10}) + T(\frac{9n}{10}) + O(n)$
(D) $T(n) = T(\frac{n}{10}) + T(\frac{9n}{10}) + O(logn)$
(E) $T(n) = 10T(\frac{n}{2}) + O(n)$

28. Minimum number of comparisons which can be done to sort an array of integers in worst case by any of the comparison based sorting algorithm is given by P while Maximum number of comparisons by any such algorithm is given by Q. Which among the following is the best approximation of $\frac{P}{Q}$:

(A) $O(1)$
(B) $O(n)$
(C) $O(\frac{1}{n^2})$
(D) $O(log_{2^n} n)$
(E) $O(\frac{logn}{nlogn})$

29. Which of the recurrence relations represent the behavior of Quick Sort in worst case?

(A)$T(n) = T(n-1) + O(n)$
(B) $T(n) = 2 * T(n-1) + O(n)$
(C) $T(n) = 2T(n/2) + O(n))$
(D) $T(n) = T(n/5) + T(4n/5) + O(n)$

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, and 4
(C) 1, 2 and 3
(D) 2, 3 and 4

31. An array with n elements has following property:
"Every element in this array is at most k distance from its position in sorted array, where k$\leq$ n."
Which sorting algorithm should be best suited for this problem?

(A) Quick sort solves this in O(knlogn)
(B) Heap Sort solves this in O(nLogk)
(C) Quick Sort solves this in O(kLogn)
(D) Merge Sort solves this in O(nLogk)
(E) Heap Sort solves this in O(kLogn)
(F) Insertion sort solves this in O(nk^2)

32. If we use a two-way merge sort algorithm to sort the following elements in ascending order 20, 47, 15, 8, 9, 4, 40, 30, 12, 17 then the order of these elements after the second pass of the algorithm is:
(A) 8, 9, 15, 20, 47, 4, 12, 17, 30, 40
(B) 8, 15, 20, 47, 4, 9, 30, 40, 12, 17
(C) 15, 20, 47, 4, 8, 9, 12, 30, 40, 17
(D) 4, 8, 9, 15, 20, 47, 12, 17, 30, 40

33. Which of the following is false about comparison based sorting algorithms?
(A) The minimum possible time complexity of a comparison based sorting algorithm is $\theta(nLogn)$ 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 algorithm

(D) Heap Sort is not a comparison based sorting algorithm

34. In a permutation a1.....an of n distinct integers, where $n > 100$, 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 $100 * n$ inversions?
(A) $\Theta(n^2)$
(B) $\Theta(nlogn)$
(C) $\Theta(n^{1.5})$
(D) $\Theta(n)$

35. If Radix sort is used to sort an array of n integers which are in the range $(n^{log_2 d}, n^{d^2})$, where d is some function of input size, the time taken would be?
(A) $O(nd^2)$
(B) $O(n^2 d \log n + n^2 \log_2 d \log_2 n)$
(C) $O(nd^2 \log_2 n)$
(D) $O(n^2 d^2 + n^2 \log_2 n)$

36. The number of elements that can be sorted in $\Theta(n2^{n \log n})$ time using merge sort is
(A) $\Theta(n^{\log n})$
(B) $\Theta((\log n)^n)$
(C) $\Theta(\frac{n \log n * 2^{n \log n}}{(\log n)})$
(D) $\Theta(\frac{2^{n \log n}}{logn})$

7 Dynamic Programming

1. If an optimal solution can be obtained for a problem by obtaining optimal solutions for its sub-problems then the problem is said to have
(A) Optimal substructure property
(B) Regular property
(C) Overlapping sub-problems property
(D) Optimal sub-problems property

2. Instead of solving a problem directly if it can be broken into many smaller sub-problems which can be reused again and again, the problem is said to have:
(A) Optimal substructure property
(B) Regular property
(C) Overlapping sub-problems property