“Insertion Sort – 1”.

1. How many passes does an insertion sort algorithm consist of?
a) N
b) N-1
c) N+1
d) N^2
View Answer
Answer: b
Explanation: An insertion algorithm consists of N-1 passes when an array of N elements is given.

2. Which of the following algorithm implementations is similar to that of an insertion sort?
a) Binary heap
b) Quick sort
c) Merge sort
d) Radix sort
View Answer
Answer: a
Explanation: Insertion sort is similar to that of a binary heap algorithm because of the use of temporary variable to swap.

3. What is the average case running time of an insertion sort algorithm?
a) O(N)
b) O(N log N)
c) O(log N)
d) $O(N^2)$
View Answer
Answer: d
Explanation: The average case analysis of a tight bound algorithm is mathematically achieved to be $O(N^2)$.

4. Any algorithm that sorts by exchanging adjacent elements require $O(N^2)$ on average.
a) True
b) False
View Answer
Answer: a
Explanation: Each swap removes only one inversion, so $O(N^2)$ swaps are required.

5. What is the average number of inversions in an array of N distinct numbers?
a) N(N-1)/4
b) N(N+1)/2
c) N(N-1)/2
d) N(N-1)/3
View Answer
Answer: a
Explanation: The total number of pairs in a list L is N(N-1)/2. Thus, an average list has half this amount, or N(N-1)/4 inversions.

6. What is the running time of an insertion sort algorithm if the input is pre-sorted?
a) $O(N^2)$
b) O(N log N)
c) O(N)
d) O(M log N)
View Answer
Answer: c
Explanation: If the input is pre-sorted, the running time is O(N), because the test in the inner for loop always fails immediately and the algorithm will run quickly.

7. What will be the number of passes to sort the elements using insertion sort?
14, 12,16, 6, 3, 10
a) 6
b) 5
c) 7
d) 1
View Answer
Answer: b
Explanation: The number of passes is given by N-1. Here, N=6. Therefore,
6-1=5 passes.

8. For the following question, how will the array elements look like after second pass?
34, 8, 64, 51, 32, 21
a) 8, 21, 32, 34, 51, 64
b) 8, 32, 34, 51, 64, 21
c) 8, 34, 51, 64, 32, 21
d) 8, 34, 64, 51, 32, 21
View Answer

Answer: d
Explanation: After swapping elements in the second pass, the array will look like, 8, 34, 64, 51, 32, 21.

9. Which of the following real time examples is based on insertion sort?
a) arranging a pack of playing cards
b) database scenarios and distributes scenarios
c) arranging books on a library shelf
d) real-time systems
View Answer
Answer: a
Explanation: Arranging a pack of cards mimics an insertion sort. Database scenario is an example for merge sort, arranging books is a stack and real-time systems uses quick sort.

10. In C, what are the basic loops required to perform an insertion sort?
a) do- while
b) if else
c) for and while
d) for and if
View Answer
Answer: c
Explanation: To perform an insertion sort, we use two basic loops- an outer for loop and an inner while loop.

11. Binary search can be used in an insertion sort algorithm to reduce the number of comparisons.
a) True
b) False
View Answer
Answer: a
Explanation: Binary search can be used in an insertion sort algorithm to reduce the number of comparisons. This is called a Binary insertion sort.

12. Which of the following options contain the correct feature of an insertion sort algorithm?
a) anti-adaptive
b) dependable
c) stable, not in-place
d) stable, adaptive
View Answer

Answer: d
Explanation: An insertion sort is stable, adaptive, in-place and incremental in nature.

13. Which of the following sorting algorithms is the fastest for sorting small arrays?
a) Quick sort
b) Insertion sort
c) Shell sort
d) Heap sort
View Answer
Answer: b
Explanation: For sorting small arrays, insertion sort runs even faster than quick sort. But, it is impractical to sort large arrays.

14. For the best case input, the running time of an insertion sort algorithm is?
a) Linear
b) Binary
c) Quadratic
d) Depends on the input
View Answer
Answer: a
Explanation: The best case input for an insertion sort algorithm runs in linear time and is given by O(N).

15. Which of the following examples represent the worst case input for an insertion sort?
a) array in sorted order
b) array sorted in reverse order
c) normal unsorted array
d) large array
View Answer
Answer: b
Explanation: The worst case input for an insertion sort algorithm will be an array sorted in reverse order and its running time is quadratic

"Insertion Sort – 2".

1. Which of the following is correct with regard to insertion sort?
a) insertion sort is stable and it sorts In-place
b) insertion sort is unstable and it sorts In-place
c) insertion sort is stable and it does not sort In-place

d) insertion sort is unstable and it does not sort In-place

View Answer

Answer: a

Explanation: During insertion sort, the relative order of elements is not changed. Therefore, it is a stable sorting algorithm. And insertion sort requires only O(1) of additional memory space. Therefore, it sorts In-place.

2. Which of the following sorting algorithm is best suited if the elements are already sorted?
a) Heap Sort
b) Quick Sort
c) Insertion Sort
d) Merge Sort

View Answer

Answer: c

Explanation: The best case running time of the insertion sort is O(n). The best case occurs when the input array is already sorted. As the elements are already sorted, only one comparison is made on each pass, so that the time required is O(n).

3. The worst case time complexity of insertion sort is $O(n^2)$. What will be the worst case time complexity of insertion sort if the correct position for inserting element is calculated using binary search?
a) O(nlogn)
b) $O(n^2)$
c) O(n)
d) O(logn)

View Answer

Answer: b

Explanation: The use of binary search reduces the time of finding the correct position from O(n) to O(logn). But the worst case of insertion sort remains $O(n^2)$ because of the series of swapping operations required for each insertion.

4. Insertion sort is an example of an incremental algorithm.
a) True
b) False

View Answer

Answer: a

Explanation: In the incremental algorithms, the complicated structure on n items is built by first building it on n − 1 items. And then we make the necessary changes to fix things in adding the last item. Insertion sort builds the sorted sequence one element at a time. Therefore, it is an example of an incremental algorithm.

5. Consider the code given below, which runs insertion sort:

```
void insertionSort(int arr[], int array_size)
{
  int i, j, value;
  for (i = 1; i < array_size; i++)
  {
          value = arr[i];
          j = i;
          while (________ )
          {
                   arr[j] = arr[j − 1];
                   j = j − 1;
          }
          arr[j] = value;
  }
}
```

Which condition will correctly implement the while loop?
a) (j > 0) || (arr[j − 1] > value)
b) (j > 0) && (arr[j − 1] > value)
c) (j > 0) && (arr[j + 1] > value)
d) (j > 0) && (arr[j + 1] < value)
View Answer
Answer: b
Explanation: In insertion sort, the element is A[j] is inserted into the correct position in the sorted sequence A[1… j – 1]. So, condition given in (j > 0) && (arr[j − 1] > value) will implement while loop correctly.

6. Which of the following is good for sorting arrays having less than 100 elements?
a) Quick Sort
b) Selection Sort
c) Merge Sort
d) Insertion Sort
View Answer

Answer: d
Explanation: The insertion sort is good for sorting small arrays. It sorts smaller arrays faster than any other sorting algorithm.

7. Consider an array of length 5, arr[5] = {9,7,4,2,1}. What are the steps of insertions done while running insertion sort on the array?
a) 7 9 4 2 1   4 7 9 2 1   2 4 7 9 1   1 2 4 7 9
b) 9 7 4 1 2   9 7 1 2 4   9 1 2 4 7   1 2 4 7 9
c) 7 4 2 1 9   4 2 1 9 7   2 1 9 7 4   1 9 7 4 2
d) 7 9 4 2 1   2 4 7 9 1   4 7 9 2 1   1 2 4 7 9
View Answer
Answer: a
Explanation: The steps performed while running insertion sort on given array are:
Initial : 9 7 4 2 1 key = 7
7 9 4 2 1 key = 4
4 7 9 2 1 key = 2
2 4 7 9 1 key = 1
1 2 4 7 9
In each step, the key is the element that is compared with the elements present at the left side to it.

8. Statement 1: In insertion sort, after m passes through the array, the first m elements are in sorted order.
Statement 2: And these elements are the m smallest elements in the array.
a) Both the statements are true
b) Statement 1 is true but statement 2 is false
c) Statement 1 is false but statement 2 is true
d) Both the statements are false
View Answer
Answer: b
Explanation: In insertion sort, after m passes through the array, the first m elements are in sorted order but they are whatever the first m elements were in the unsorted array.

9. In insertion sort, the average number of comparisons required to place the 7th element into its correct position is ____
a) 9
b) 4
c) 7
d) 14
View Answer

Answer: b
Explanation: On average (k + 1) / 2 comparisons are required to place the kth element into its correct position. Therefore, average number of comparisons required for 7th element = (7 + 1)/2 = 4.

10. Which of the following is not an exchange sort?
a) Bubble Sort
b) Quick Sort
c) Partition-exchange Sort
d) Insertion Sort
View Answer
Answer: d
Explanation: In Exchange sorts, we compare each element of an array and swap those elements that are not in their proper position. Bubble Sort and Quick Sort are exchange sorts. Quick Sort is also called as Partition-exchange Sort. Insertion sort is not an exchange sort.