**1. General Definitions & Concepts**

* What is sorting? Define sorting.
* Differentiate between internal sorting and external sorting. Why is external sorting slower than internal sorting? Why do you need external sorting?
* What do you mean by adaptive and non-adaptive sorting?

**2. Sorting Algorithms: Time Complexity & Efficiency**

* What is the worst-case time complexity of quicksort?
* Define stability and efficiency of sorting algorithms. Compare sequential search with binary search.

**3. Specific Sorting Algorithms**

* Explain selection sort.  
  Sort the numbers 92, 73, 22, 49, 46, 98, 21, 9, 70, 63 using selection sort.
* What is insertion sort? Perform insertion sort on the following data:  
  14, 33, 27, 10, 35, 19, 42, 44.
* Define binary search with an example.
* What is internal sorting?
* Sort the following data using the quicksort algorithm:  
  20, 50, 45, 36, 8, 90, 85, 34.

**4. Practical Application of Sorting Algorithms**

* Sort the following data using the insertion sort algorithm in ascending order:  
  14, 33, 27, 10, 35, 19, 42, 44.
* Show different passes required to perform selection sort on the following list of numbers:  
  76, 32, 43, 10, 87, 21, 65, 54.
* Sort the following data using radix sort.
* Write a recursive algorithm for quicksort and trace the algorithm for the following data:  
  35, 22, 10, 3, 48, 29, 6, 42, 8, 75.
* Sort the following elements using a max heap and also analyze its computational complexity:  
  42, 36, 56, 27, 63, 72, 62, 15.

**5. Heap and Binary Search Tree**

* Consider the following max heap: 50, 30, 20, 15, 10, 8, 16. Insert a new node with value 60.

**1. Shell Sort**

* Explain shell sort. Sort the numbers 92, 83, 22, 49, 36, 98, 12, 9, 70, 51 using shell sort.
* What is internal and external sorting? Write an algorithm for shell sort.

**2. Selection Sort**

* Explain selection sort. Sort the data sequence 40, 90, 20, -10, 30, 5, 60, 100, 80 using selection sort.

**3. Heap Sort**

* Construct a heap sort for the following given list with an algorithm: 37, 33, 26, 92, 57, 18, 48, 25, 12, 86, 42, 22.
* Create the heap structure from the following sequence data and sort them using heap sort: 12, 10, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, 2.

**4. Binary Search and Hashing**

* Explain a binary search with an example.
* Consider a hash table of size 10. Using linear probing, insert the keys 62, 37, 36, 44, 67, 91, 82, and 107.

**5. Radix Sort**

* Define a radix sort with its algorithm.
* Explain the basic principle of radix sort. Trace the sorting steps in quick sort and radix sort. 25, 57, 48, 37, 12, 92, 86, 33.

**6. Merge Sort**

* Trace the steps to sort the following set of data using merge sort: 85, 76, 46, 92, 30, 41, 12.

**7. Comparison of Sorting Algorithms**

* How can you compare either two sorting or two searching algorithms?
* What are the conflicting efficiency considerations in various sorting methods?
* Compare and contrast the efficiency of Bubble sort, Quick sort, Insertion sort and Selection sort algorithms with an example.
* Conflicting efficiency considerations in various sorting methods?

**8. Quick Sort**

* Write an algorithm for quick sort and trace your algorithm for a given sequence of data 5, 43, 99, 20, 45, 7, 6, 63, 92, 4.
* Write the algorithm of the quick sort including the steps of partition. Discuss the complexity of this algorithm.
* Explain the basic principle of radix sort. Trace the sorting steps in quick sort and radix sort. 25, 57, 48, 37, 12, 92, 86, 33.

**9. Big-O and Time Complexities**

* Provide the best case, average case, and worst case for the following algorithms in Big-Oh: bubble sort, insertion sort, merge sort, and selection sort.
* Define Big 'O' notation. Compare linear logarithmic, linear, and quadratic order function. Explain which elementary sorting algorithm (i.e. Bubble, Insertion, Selection) you choose when the input data is in almost sorted form.

**10. Internal and External Sorting**

* Define internal and external sorting.

**Heap**

1. What is a heap?

**Quick Sort**

1. Explain quick sort algorithm with Big-oh notation in best case, average case, and worst case and trace it to sort the data: 8, 10, 5, 12, 14, 5, 7, 13.

**Binary Search**

1. What is binary search? Write an algorithm to search an item using binary search.
2. Explain the binary search algorithm. Illustrate it with an example.
3. Trace Binary Search algorithm for the data: 21, 36, 56, 79, 101, 123, 142, 203 and Search for the values 123 and 153.
4. Compare and contrast binary search and binary tree search.
5. Discuss the binary search technique along with its efficiency.

**Sorting: General**

1. What is sorting?
2. Write short notes on (any two): External Sorting.
3. What are external and internal sorting? Explain partition strategies of Merge sort and Quick sort.
4. Compare the partition strategies of Merge sort and Quick sort.

**Insertion Sort**

1. What is insertion sort? Trace and sort the following data using insertion sort: 90, 57, 80, 10, 22, 21, 45, 9, 78.
2. Hand test the insertion sort algorithm with the following array of numbers: 16, 7, 31, 2, 9, 41, -10.

**Merge Sort**

1. Discuss merge sort. How do you rate this sorting compared to selection sort?
2. Write an algorithm and C function for merge sort.
3. Trace Merge sort for the following data: 11, 45, 61, 33, 55, 9, 83, 25.

**Selection Sort**

1. Write a program to sort an array using selection sort.
2. Differentiate between selection sort and bubble sort.

**Bubble Sort**

1. Explain the Bubble sort algorithm. Illustrate it with an example.
2. Write a program in C for bubble sorting.

**Sorting Trace Example**

1. Trace these sort algorithms for the following data: 11, 45, 61, 33, 55, 9, 83, 25.