UNIT III – Question Bank: Sorting, Searching & Hashing

Part A – 2 Marks (Short Answer Questions)

1. 1. Define bubble sort with an example.
2. 2. What is the basic idea of selection sort?
3. 3. List the time complexities of insertion sort in best and worst cases.
4. 4. What is divide and conquer approach?
5. 5. Write the worst-case time complexity of quick sort.
6. 6. What is the difference between linear search and binary search?
7. 7. Define hash function. Give one example.
8. 8. What is a collision in hashing?
9. 9. Explain load factor in hashing.
10. 10. Define rehashing. When is it necessary?
11. 11. What is the time complexity of binary search in a sorted array?
12. 12. List two applications of hashing.
13. 13. What is meant by stability in sorting algorithms?
14. 14. What are the advantages of merge sort over bubble sort?
15. 15. Define open addressing in collision resolution.

Part B – 13 Marks (Descriptive/Problem Solving Questions)

16. 1. Explain the working of bubble sort with a sample input. Derive its time complexity.
17. 2. Write and explain the algorithm for selection sort. Show its execution on a sample list.
18. 3. Discuss insertion sort with a suitable example. Analyze its best and worst case time complexities.
19. 4. Explain merge sort algorithm using divide and conquer strategy. Analyze its time and space complexity.
20. 5. Describe the quick sort algorithm with example and analyze its time complexity in all cases.
21. 6. Compare merge sort and quick sort in terms of efficiency, use case, and space requirements.
22. 7. Write an algorithm for binary search and trace it for a sorted list.
23. 8. Explain linear search with an example and compare its efficiency with binary search.
24. 9. Describe different hash functions and explain how they affect performance.
25. 10. What are the different collision handling techniques in hashing? Explain with examples.
26. 11. Explain load factor and rehashing. Why are they important in hash table performance?
27. 12. Compare and contrast open addressing and chaining in hash tables.

28. 13. Design a hash table using division method and demonstrate insertion and collision handling using chaining.

Part C – 15 Marks (Case Study / Application-Based Questions)

29. 1. Student Records Sorting System: You are developing a system that sorts student records based on marks. Choose and justify the best sorting algorithm (merge sort or quick sort). Demonstrate with sample data and provide code.
30. 2. Library Book Search: Design a system that uses binary search to find books in a sorted database of titles. Explain how it works, and compare it with linear search in performance.
31. 3. Online Dictionary Lookup: Implement a search mechanism for words using hashing. Explain the selection of hash function and collision handling. Provide a working example with word insertions and lookups.
32. 4. Medical Report Indexing: Patient records need to be accessed using unique IDs. Design a hashing-based lookup system to support fast retrieval. Include load factor and rehashing in your explanation.
33. 5. Sorting Evaluation Tool: Build a tool that benchmarks bubble, selection, and insertion sort on the same dataset. Analyze their performance on best, average, and worst cases.
34. 6. University Database Management: A university stores thousands of student records. Propose a combined system using hashing for ID lookup and quick sort for sorting by name. Justify your choices and implementation.