**1. Definition and Basic Concepts of Hashing:**

* What is a hash function?
* What is hashing?
* What is open hashing?
* What do you mean by open hashing and closed hashing?
* What do you mean by rehashing and double hashing?
* Define Hash collision and collision resolution.
* What is hashing? Why do we need hashing? Give the characteristics of a hash function.
* Differentiate between hashing and sequential search.
* What is hashing? What is a collision? What are the techniques used for collision resolution in hashing?

**2. Collision and Collision Resolution Techniques:**

* What is collision? What are the techniques used for collision resolution in hashing?
* What are the different collision reduction techniques in hashing?
* Briefly explain different types of collision resolution techniques.
* How can the collision in hashing be reduced? Explain any one collision reduction technique.
* Discuss linear probing in detail.
* What is rehashing and double hashing?

**3. Hash Table Construction and Probing Techniques:**

* Given a set of elements A = {3, 2, 9, 6, 11, 9, 12} and hash functions h(k) = 2k + 3 and h2(k) = 3k + 1, insert these elements into a hash table of size 10 using the division method and double hashing technique for open addressing.
* Construct a hash table of size 10 for the following set of data using linear probing, quadratic probing, and chaining with hash function X mod 10: 21, 36, 39, 42, 44, 46, 55, 66, 91, 35.
* Given input {3, 2, 9, 6, 11, 13, 7, 2} and a hash function h(k) = 2k + 3. Show the resulting hash table using i. Linear Probing ii. Quadratic Probing.
* Insert the keys: 62, 37, 36, 44, 67, 91, 82, and 31 using the quadratic probing method. The hash function is h(key) = key % 10 (where 10 is the table size).
* Consider a hash table of size 10. Using quadratic probing, insert the keys 52, 77, 26, 44, 33, 91, and 81 into the table. (Take C1 = 1 and C2 = 3).

**1. Hashing and Collision:**

* Define hashing and hash collision.
* What is the cause of collision in hashing and explain any one method for the collision resolution?
* Define hashing and hash collision. How do you minimize the hash collision?

**2. Collision and Collision Resolution:**

* What is collision? Explain any two methods of collision resolution with reference to hashing.
* Explain with example the collision resolution method open hashing.
* Insert elements into a hash table using modulo division method with double hashing.
* Discuss about linear probing and quadratic probing.
* How are linear probing, quadratic probing, and double hashing techniques used to resolve a collision? Explain with a suitable example.
* Explain any three collision resolving schemes with examples.

**3. Clustering:**

* What is clustering? Explain any three collision resolving schemes with examples.
* Define clustering in rehashing method. Is it possible to remove clustering by quadratic probing? Explain.

**4. Probing Techniques and Algorithms:**

* Insert the keys 30, 15, 69, 28, 49, 58, 79, and 91 using the quadratic probing method. The hash function is: h(key) = key % 10.
* Write an algorithm for collision resolution by open addressing.
* Outline an algorithm to delete a key from a hash table when the linear probing is used for inserting keys.

**5. Binary Search and Other Hash Table Operations:**

* Write binary search. Consider a hash table of size 10; insert the keys 62, 37, 36, 44, 67, 91 and 107 using linear probing.

**6. Serial vs Parallel Algorithm:**

* Write the difference between serial and parallel algorithm with example.

**Hashing Concepts:**

1. Write short notes on a) Hashing.
2. What is hashing? Discuss rehashing with an example.
3. What are Hashing and collision? Write about any three hashing algorithms.
4. Why do we need Hashing? Discuss linear probing in detail.

**Hash Functions and Collision Handling:**

1. Hash function.
2. State collision resolution techniques in hashing. Explain double hashing and quadratic probing techniques.

**Big O Notation and Algorithm Analysis:**

1. What is Big 'O' notation? Analyze anyone sorting algorithm.