FAQ

Module-5	
Question 1.	**What is the fundamental difference between HashTables and HashMaps?**
Answer	HashTables are known for their synchronisation and thread-safety properties, which contribute to their somewhat slower performance compared to HashMaps. In contrast, HashMaps lack synchronisation and thread safety, hence providing enhanced efficiency in single-threaded contexts.
Question 2.	**In what ways do Binary Trees exhibit distinctions from Binary SearchTrees (BSTs)?**
Answer	Binary trees are a kind of hierarchical data structure that consist of nodes, each of which may have a maximum of two offspring. These trees do notimpose any restrictions on the values of the nodes. Binary Search Trees (BSTs) are a kind of binary tree that exhibit an ordered or sorted structure. In a BST, the left child node is assigned a value that is less than its parent node, while the right child node is assigned a

	value that is higher than its parent node.
Question 3.	**What are the advantages of using an AVL Tree over a standard BinarySearch Tree (BST)?**
Answer	AVL Trees, a variant of Binary Search Trees designed for self-balancing, guarantee the preservation of balance in the tree after each insertion or deletion operation. The presence of balance in trees ensures that all tree operations have logarithmic limitations, hence preventing the occurrence of worst-case situations seen in binary search trees, where they may deteriorateinto linked lists.
Question 4.	**What is the underlying objective of using Priority Queues?**
Answer	Priority queues are a kind of data structure that organises componentsaccording to their respective priority. These data structures enable the retrieval of the highest (or lowest) priority element in constant time, irrespective of the sequence in which elements were inserted.

Question 5.	**In which situations would Heaps be considered an optimal choice for adata structure?**
Answer	Heaps are well-suited for use in applications that need regular management ofan item with the greatest or lowest priority. Common applications of this include the implementation of algorithms like Dijkstra's shortest route algorithm, Prim's Minimum Spanning Tree method, or any situation that necessitates the efficient retrieval of the highest or lowest value, such as inPriority Queues.