**Changes in HashMap Implementation in Java 8 (Handling Hash Collisions Efficiently)**

In **Java 7 and earlier**, when multiple keys map to the same bucket (due to hash collisions), **entries were stored as a linked list inside the bucket**.

However, **Java 8 introduced a major optimization**:  
If a bucket contains **many entries (i.e., high collision rate)**, the linked list is **converted into a balanced binary search tree (BST)** for better performance.

**Why Was This Change Introduced?**

* In Java 7, when multiple keys collide and fall into the same bucket, they form a **linked list**.
* Searching for a key in a **linked list takes O(n) time** in the worst case.
* Java 8 **replaces the linked list with a balanced tree** (TreeNode) **if the number of elements in a bucket exceeds a threshold (8)**.
* **Tree-based lookups take O(log n) time**, which improves performance for large HashMaps.