# java tree map – Assessment Bank

Generated on: 2025-07-19 07:21:49

1. A software engineer needs to store and retrieve student records (identified by their unique ID numbers) efficiently, prioritizing fast retrieval times. Which of the following statements best evaluates the suitability of using a TreeMap over a HashMap for this task?  
  
 A. A HashMap is always preferable due to its faster average-case performance.  
 B. A TreeMap offers slightly faster retrieval, but at the cost of increased memory usage, making it unsuitable for large datasets.  
 C. A TreeMap is the better choice because its sorted key-value pairs allow for efficient retrieval based on ID number, even with a large number of records. This is critical if there's a need for ordered access.  
 D. Both TreeMap and HashMap are equally suitable; the choice depends solely on personal preference.  
 \*\*(Answer: C)\*\*  
  
  
2. Consider a scenario where you need to maintain a sorted list of words and their frequencies. Evaluating the performance implications, which data structure offers the best balance between efficient insertion, retrieval, and maintaining sorted order? Why?  
  
 A. ArrayList  
 B. LinkedList  
 C. HashMap  
 D. TreeMap  
 \*\*(Answer: D)\*\*  
  
  
3. You're tasked with optimizing a Java application that uses a TreeMap to store a large number of key-value pairs. Performance testing reveals slowdowns during frequent insertion operations. Which of the following modifications is LEAST likely to significantly improve insertion performance?  
  
 A. Implementing a custom Comparator for keys to optimize the comparison process.  
 B. Using a different data structure entirely, such as a HashMap.  
 C. Switching to a concurrent TreeMap implementation for multi-threaded scenarios.  
 D. Optimizing the key's `hashCode()` and `equals()` methods if keys are custom objects.  
 \*\*(Answer: A)\*\*  
  
  
4. A program uses a TreeMap to store data. Considering potential limitations, which of the following scenarios BEST illustrates a situation where using a TreeMap might be inefficient compared to other data structures?  
  
 A. Storing and retrieving student names alphabetically sorted by last name.  
 B. Maintaining a list of frequently accessed items for a cache.  
 C. Implementing an in-memory database for quick lookups of user data based on a unique ID.  
 D. Storing and retrieving data with frequent insertions and deletions where the order doesn't matter.  
 \*\*(Answer: D)\*\*  
  
  
5. Which statement accurately evaluates the use of a TreeMap's `subMap()` method in terms of performance and practicality?  
  
 A. `subMap()` provides a constant-time operation, irrespective of the sub-map size.  
 B. `subMap()` is generally inefficient for extracting small sub-sections of the map.  
 C. `subMap()` offers a practical approach to efficiently access and manipulate portions of a large TreeMap without copying data.  
 D. `subMap()` provides a complete copy of the selected portion, hence offering no performance advantages.  
 \*\*(Answer: C)\*\*  
  
  
6. Evaluating the complexity of TreeMap operations, which statement is most accurate regarding the time complexity of the `put()` method?  
  
 A. O(1)  
 B. O(n)  
 C. O(log n)  
 D. O(n^2)  
 \*\*(Answer: C)\*\*  
  
  
7. Considering memory usage, how does a TreeMap compare to a HashMap when storing a large number of entries?  
  
 A. A TreeMap generally consumes less memory than a HashMap.  
 B. A HashMap always consumes less memory than a TreeMap.  
 C. Memory consumption is roughly the same for both.  
 D. Memory consumption depends heavily on the size of the keys and values, regardless of the data structure used.  
 \*\*(Answer: D)\*\*  
  
  
8. You need to implement a system that keeps track of real-time stock prices, ordered by stock symbol. Evaluating the different tree-based data structures in Java, which would be MOST suitable considering the need for efficient insertion and retrieval of updates?  
  
 A. TreeSet  
 B. TreeMap  
 C. PriorityQueue  
 D. BinarySearchTree (not provided in the standard Java library)  
 \*\*(Answer: B)\*\*  
  
  
9. What is a potential disadvantage of using a TreeMap when compared to a HashMap in a high-concurrency environment?  
  
 A. TreeMap offers better thread safety than HashMap.  
 B. TreeMap is inherently faster than HashMap for concurrent access.  
 C. TreeMap’s synchronized methods can lead to performance bottlenecks during heavy contention.  
 D. TreeMap's performance is unaffected by concurrent access.  
 \*\*(Answer: C)\*\*  
  
  
10. A developer is choosing between a TreeMap and a HashMap for a project. Which factor would MOST strongly favor selecting a TreeMap over a HashMap?  
  
 A. The need for extremely fast average-case insertion speeds.  
 B. The requirement for unsorted key-value pairs.  
 C. The need for keys to be maintained in a specific sorted order.  
 D. The system needs to handle a very high volume of concurrent read/write operations.  
 \*\*(Answer: C)\*\*

---  
Watermark: TeachMate AI | Version 1.0