HashSet vs TreeSet in Java

Difference between SJF and LJF CPU scheduling algorithms

Difference between Iterators and Pointers in C/C++ with Examples

exit(0) vs exit(1) in C/C++ with Examples

Difference between DELETE and TRUNCATE

Difference between CSMA/CA and CSMA/CD

Difference between Left, Right and Full Outer Join

Monolithic vs Microservices

architecture

Difference between SCAN and CSCAN Disk scheduling algorithms

Difference between Virtual function and Pure virtual function in C++

Difference between Prim's and Kruskal's algorithm for MST

Difference between 2NF and 3NF in DBMS

Difference between Increment and Decrement Operators

Difference between Core Java and Advanced Java

Difference between Argument and Parameter in C/C++ with Examples

Difference between order by and group by clause in SQL

Difference between Static variables and Register variables in C

What are the differences between Bellman Ford's and Dijkstra's algorithms?

Difference between Type Casting and Type Conversion

Difference between JOIN and UNION in SQL

Difference between MySQL and PostgreSQL

Difference between E-R Model and Relational Model in DBMS

Internal static variable vs. External static variable with Examples in C

Difference between Pushdown Automata and Finite Automata

Difference between Clustered and Non-clustered index

Difference between URL and URI

Difference between Contiguous and Noncontiguous Memory Allocation

Difference between Binary tree and B-tree

Difference between Static Allocation and Heap Allocation

Difference between Trigger and Procedure in DBMS

HashSet vs TreeSet in Java

- **Speed and internal implementation**

  HashSet : For operations like search, insert and delete. It takes constant time for these operations on average. HashSet is faster than TreeSet. HashSet is Implemented using a hash table.

  TreeSet : TreeSet takes O(Log n) for search, insert and delete which is higher than HashSet. But TreeSet keeps sorted data. Also, it supports operations like higher() (Returns least higher element), floor(), ceiling(), etc. These operations are also O(Log n) in TreeSet and not supported in HashSet. TreeSet is implemented using a Self Balancing Binary Search Tree (Red-Black Tree). TreeSet is backed by TreeMap in Java.

- **Ordering**
  Elements in HashSet are not ordered. TreeSet maintains objects in Sorted order defined by either Comparable or Comparator method in Java. TreeSet elements are sorted in ascending order by default. It offers several methods to deal with the ordered set like first(), last(), headSet(), tailSet(), etc.
- **Null Object**
  HashSet allows null object. TreeSet doesn’t allow null Object and throw NullPointerException, Why, because TreeSet uses compareTo() method to compare keys and compareTo() will throw java.lang.NullPointerException.
- **Comparison**
  HashSet uses equals() method to compare two object in Set and for detecting duplicates. TreeSet uses compareTo() method for same purpose.
  If equals() and compareTo() are not consistent, i.e. for two equal object equals should return true while compareTo() should return zero, than it will break contract of Set interface and will allow duplicates in Set implementations like TreeSet

If you want a sorted Set then it is better to add elements to HashSet and then convert it into TreeSet rather than creating a TreeSet and adding elements to it.

HashSet example

```java
// Java program to demonstrate working of
// HashSet
import java.util.HashSet;
class HashSetDemo {
    public static void main(String[] args)
    {

        // Create a HashSet
        HashSet<String> hset = new HashSet<String>();

        // add elements to HashSet
        hset.add("geeks");
        hset.add("for");
        hset.add("practice");
        hset.add("contribute");

        // Duplicate removed
        hset.add("geeks");

        // Displaying HashSet elements
        System.out.println("HashSet contains: ");
        for (String temp : hset) {
            System.out.println(temp);
        }
    }
}
```

Output:

```
HashSet contains:
practice
geeks
for
contribute
```

TreeSet example

```java
// Java program to demonstrate working of
// TreeSet.
import java.util.TreeSet;
class TreeSetDemo {

    public static void main(String[] args)
    {
        // Create a TreeSet
        TreeSet<String> tset = new TreeSet<String>();

        // add elements to HashSet
        tset.add("geeks");
        tset.add("for");
        tset.add("practice");
        tset.add("contribute");

        // Duplicate removed
        tset.add("geeks");

        // Displaying TreeSet elements
        System.out.println("TreeSet contains: ");
        for (String temp : tset) {
            System.out.println(temp);
        }
    }
}
```

Output:

```
TreeSet contains:
contribute
for
geeks
practice
```

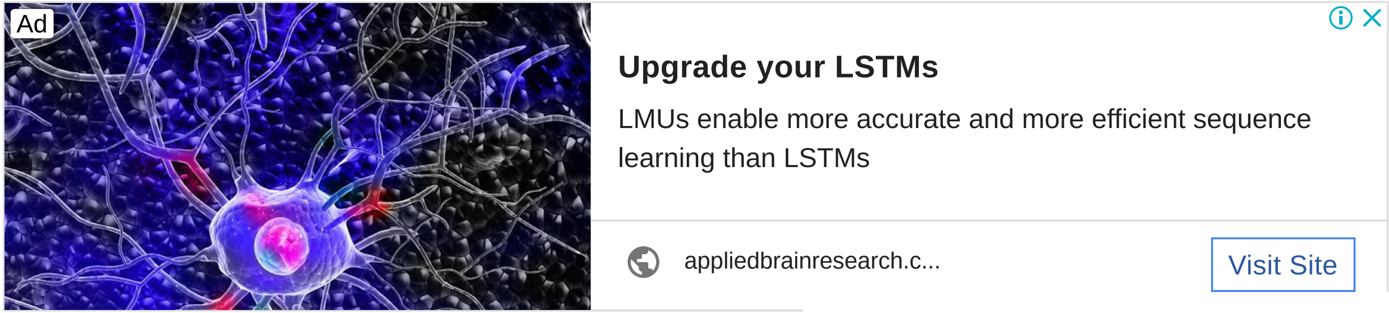
When to prefer TreeSet over HashSet

1. Sorted unique elements are required instead of unique elements. The sorted list given by TreeSet is always in ascending order.

2. TreeSet has greater locality than HashSet.If two entries are near by in the order, then TreeSet places them near each other in data structure and hence in memory, while HashSet spreads the entries all over memory regardless of the keys they are associated to.

3. TreeSet uses Red- Black tree algorithm underneath to sort out the elements. When one need to perform read/write operations frequently, then TreeSet is a good choice.

4. LinkedHashSet is another data structure that is between these two. It provides time complexities like HashSet and maintains order of insertion (Note that this is not sorted order, but the order in which elements are inserted).



Recommended Posts:

Convert HashSet to TreeSet in Java

Difference and similarities between HashSet, LinkedHashSet and TreeSet in Java

HashSet in Java

HashSet add() Method in Java

How to sort HashSet in Java

HashSet contains() Method in Java

Traverse through a HashSet in Java

Initializing HashSet in Java

HashSet equals() method in Java with Example

HashSet toArray() method in Java with Example

Difference between ArrayList and HashSet in Java

HashSet containsAll() method in Java with Example

HashSet retainAll() method in Java with Example

HashSet clear() Method in Java

HashSet toString() method in Java with Example

knowledgeenlightens22
Check out this Author's contributed articles.

Article Tags : Difference Between Hash Java Tree Java-Collections java-hashset Java-Set-Programs java-treeset

Practice Tags : Hash Java Tree Java-Collections

👍
3

2

□ To-do □ Done

Based on **1** vote(s)

Feedback/ Suggest Improvement Add Notes Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY
About Us
Careers
Privacy Policy
Contact Us

LEARN
Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE
Courses
Company-wise
Topic-wise
How to begin?

CONTRIBUTE
Write an Article
Write Interview Experience
Internships
Videos