

Array, Set, Map 的多种实现

2016-12-15

Java

Java 作为一门发展多年的成熟的编程语言，提供了丰富的基础数据结构。即使是简单的数组 Array，集合 Set，字典 Map，Java 中都有多种实现。

这种在 Swift 这种语言中就看不到。Swift 中，你只有一个 Array，一个 Set，一个 Dictionary。如果，你想要一个 linked list，一个 sorted set，或者一个 sorted dictionary。对不起，你需要自己实现。

Java 对各种数据结构做了更细致的区分。这样程序员可以为不同的场景选择最合适的数据结构，以编写出性能更好的代码。这是一件好事。当然，这也增加了这门语言的学习成本。Java 程序员需要理解清楚各种 Array，Set，Map 的各个具体实现之间的差别，才能很好地应用它们。

Array

LinkedList

LinkedList，即链表。拥有很好的插入，和删除性能，这两个操作的时间复杂度都是： $O(1)$ 。但是，链表的查询，更新操作性能较差。这两个操作的时间复杂度是： $O(n)$ 。这是因为，对链表元素的查询是需要做线性查找的。

ArrayList

ArrayList，和 Java 最基础的 Array 相比，拥有自增减容量功能。Array List 每次自增 50%。插入，删除操作较链表要慢，这两个操作的时间复杂度： $O(n)$ 。但是，对于首元素，和尾元素做插入，和删除操作，Java 做了优化时间复杂度为 $O(1)$ 。

查询，更新操作较链表要好，这两个操作的时间复杂度： $O(1)$ 。

Vector

Vector 和 ArrayList 是类似的。

而它们最重要的区别在于，Vector 是 Synchronized，也就是说，它的插入，删除和更改操作是线程安全的。这一点 ArrayList 并不具备。

Array 的选择

- 很多的插入，和删除，但没有太多的随机查询的情况下，使用 LinkedList。
- 有很多随机查询，使用 ArrayList。
- 有线程安全要求，使用 Vector。

Set

集合 Set，是一组无重复的元素。

HashSet

HashSet 以哈希表实现。插入的元素是无序的。add, remove 和 contains 都是常数时间复杂度 $O(1)$ 。

TreeSet

TreeSet 以红黑树实现。插入的元素由树结构保持了顺序。add, remove 和 contains 都是对数时间复杂度 $O(\log n)$ 。但是，它带来了查找 search，以及有序操作 first, last, headSet, tailSet 的高效。

LinkedHashSet

LinkedHashSet 是在 HashSet 基础上，为元素添加了双向指针，以形成双向链表。这样，元素的插入顺序得以保存。但由于元素的顺序是以链表形式保存的。其查找操作，只能是线性查找，时间复杂度是 $O(n)$ 。这没有以树结构实现的 TreeSet 高效，TreeSet

的查找操作时间复杂度是 $O(\log n)$ 。这种链表形式可以提供简单的顺序操作，如按插入顺序输出所有元素。

Set 的选择

- 无顺序操作要求，使用 HashSet。
- 有简单顺序操作要求，如按插入顺序输出元素，使用 LinkedHashSet。
- 有复杂顺序操作要求，如区间搜索，即返回一个区间的所有元素，使用 TreeSet。

Map

字典 Map，是一个符号表的典型实现。符号表建立了标示符到其声明值的对应关系。符号表有多种实现。

字典 Map 是符号表最常见的实现。字典 Map 这一组实现上和集合 Set 有类似的地方。实现它们的基础数据结构都是哈希表，红黑树，链表。所以，它们的特点也呈现出类似的对比关系。

HashMap

HashMap 是通过哈希表实现的，其键是无序的。

Hashtable

Hashtable 和 HashMap 类似。但 Hashtable 是 synchronized，线程安全，但这也带来了额外的开销。

TreeMap

TreeMap 以红黑树实现，其键是有序的。树结构，使得可以对键做比较高效的查找，插入操作。以及有序操作。如 first，tail 等。

LinkedHashMap

LinkedHashMap 通过在键之间形成双向链表而保持了插入顺序。这种链表形式可以提供简单的顺序操作，如按插入顺序输出所有键。

Map 的选择

- 无顺序操作要求，使用 HashMap。
- 有线程安全要求，使用 Hashtable。
- 有简单顺序操作要求，如按插入顺序输出键，使用 LinkedHashMap。
- 有复杂顺序操作要求，如区间搜索，即返回一个区间的所有键，使用 TreeMap。

参考链接

- [ArrayList vs. LinkedList vs. Vector](#)
- [Difference between ArrayList and Vector In java](#)
- [HashSet vs. TreeSet vs. LinkedHashSet](#)
- [HashMap vs. TreeMap vs. Hashtable vs. LinkedHashMap](#)

[点击查看评论](#)