1.说说 List,Set,Map 三者的区别？三者底层的数据结构？

List的本质是列表，Map的本质是数组加链表，Set的本质是Map的Key 集合

List是集合包括链表和数组，数组需要向内存申请一段连续的空间ArrayList的初始大小是10，扩容1.5,。链表因为查找下一个数据依靠Node，所以不需要一段连续的空间也可以存储，也不需要提前分配内存空间，集合的话查找比较好，链表的话增删比较好

Map的本质是通过数组加链表来实现，Map基本用到的有HashMap、HashTable和ConcurrentHashMap，HashMap是线程不安全的但是效率高，可以接受null，在单线程的情况下一般都使用HashMap，初始值16，两倍扩容。HashTable是线程 安全的，通过将所有方法加上Synchronize保证线程安全，但是由于效率较低一般不使用。ConcurrentHashMap是通过在之前是通过模块加锁的方式来时先线程安全，在之后的jdk对其进行了优化，才用了Node节点+CAS的方式完成加锁提高了效率

Set的实现原理很简单，就是利用了Map中Key不能重复的实现，来实现了Set去重的操作

2.有哪些集合是线程不安全的？怎么解决呢？

ArrayList,HashMap这些都是线程不安全的，java通过封装他们的线程安全类如：vector，HashTable 。 他们都是通过所有方法加synchronize 来实现线程安全，所有普遍效率较低。

3.比较 HashSet、LinkedHashSet 和 TreeSet 三者的异同

三者由于都是Set所以值都是不能重复的，HashSet数据是无序的存入和取出的顺序不同，LinkedHashSet运用了链表的思想，存入时候的数据和取出时候的数据顺序相同，TreeSet是排序Set能够根据一定的规则，也可以自定义规则来实现排序，只需要让需要排序的实体类实现Comparable ，重写他的CompareTo方法就行了

4.HashMap 和 Hashtable 的区别？HashMap 和 HashSet 区别？HashMap 和 TreeMap 区别？

HashMap和Hashtable都是Map，区别在于Hashtable是在HashMap的基础上实现了线程安全，也就是将所有的方法都加上了sychronized，让所以方法变成线程安全。

HashMap和HashSet区别在于HashSet是根据HashMap来实现的HashSet存储值的时候是用HashMap计算key时候的hashcode来存储，保证数据的唯一性。

HashMap和TreeMap的区别是TreeMap加入了排序操作，可以通过实现comparable重写compareTo方法来实现。

5.HashMap 的底层实现

HashMap是通过数组+链表的方式实现的，在1.8之后引入了红黑树的概念，即当链表的长度达到8的时候链表就会转换为红黑树，数组则是通过对存入对象Hashcode进行计算当Hashcode相同时，则存储在同一个链表中，HashMap的初始容量大小为16，每次两倍扩容，其中涉及到Hash算法。

6.ConcurrentHashMap 和 Hashtable 的区别？ConcurrentHashMap 线程安全的具体实现方式/底层具体实现

Hashtable为保证线程安全时通过同一把锁的方式，当一个方法在调用某个数据时，其他的数据也都无法进行调用，所以说效率不高。

ConcurrentHashMap不同于Hashtable，在1.7之前，concurrentHashMap是通过分段锁的方式对数据进行加锁，在1.8之后根据最新的HashMap进行改变，只对采用CAS和node节点进行加锁，只有相同的hashcode进行操作时，才会进行锁操作，提高效率。

7.HashMap 的长度为什么是 2 的幂次方

先看看HashMap存储位置的计算方法，h&(lenth-1)，计算时发现-1操作不影响操作结果，但是会让计算更快。而与操作让h在进行操作时，如果lenth是偶数，则-1后是奇数，在进行&操作时可以得到偶数，也可以得到奇数，但是如果是奇数则只能得到偶数。而通过计算可以发现，2的幂次方可以更加减少key之间的碰撞，从而加快查询效率。因为hashmap计算code 的时候是取低位进行操作，而&操作最好就是低位全是1，那操作才有意义，所以使用2的幂次方

8.ArrayLis线程安全

Vectory线程安全，两倍扩容，通过对所有方法进行加锁来实现线程安全。

CopyOnWriteArrayList通过cow思想来实现线程安全，在add方法中先copy旧数组出一个新数组，然后将add的数据add到新数组中，然后再把array的指向指到新数组中。cow只能保证数据的最终一致性，也就是说a线程读取数据的时候，b现场将数据清空了，a线程还是可以读取到数据。