java ArrayList与LinkedList，jDK7

资料一直都说arrylist查找会比linkedlist快，linkedlist增删比arrayList快，但我测试结果却相反，难道是我代码写错了，代码如下：  
//List l = new ArrayList（）;  
List l = new LinkedList();  
  
//time 1  
for(int y=0;y<1000000;1++)  
l.add(i);  
  
//time2  
Iterator i = l.iterator();  
while(i.hasNext()){  
System.out.println(i.next());  
  
}  
  
//time3  
  
我是分开6次测试，分三组，每组5次，分别是：  
time3-time2：5次平均值linkedList胜利,A:498.6;L329  
time2-time1：5次平均值A:118.6，L:246.6  
time3-time1：5次平均值ArrayList胜利A505.4,L548.8  
求解释

每组测试，我都会将不需要的屏蔽，例如time2-time1不需要while，我会屏蔽它，不会有多余的代码

2013-05-23 16:21

[＃“谁是脑力达人”赛前预热正式开启！＃](http://zhidao.baidu.com/activity/commact?name=nldryr&fr=wzl)

提问者采纳

1. 单纯的add在最后面加, 我觉得两者没有什么区别, 谁快谁慢都有可能, 甚至linkedList可能会慢, 因为它要先从头link到尾, 再在尾加, 而arraylist直接可以定位到尾部, 或者linkedlist也有尾指针, 那样直接加也可以..所以看具体实现了 //JDK1.7LinkedList有尾指针  
2. arrayList定位快是毋庸置疑的, 直接通过下标定位的方式任何位置都是常数时间, 而linkedlist则要通过移动指针  
3. linkedlist添加删除快, 是因为只需要操作指针指向就可以了, 而arraylist则还要移动数据, 比如添加时, 后面的数据都要向后移从而腾出空间

追问：

那我要测试出概念上的说法应该怎么测试呢？

追答：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | import java.util.ArrayList;  import java.util.LinkedList;  import java.util.List;  public class Test031 {      /\*\*       \* @param args       \*/      public static void main(String[] args) {          // TODO Auto-generated method stub          List<String> list1 = new ArrayList<String>();          List<String> list2 = new LinkedList<String>();          for (int i = 0; i < 100000; i++) {              if (i < 10) {                  list1.add("a");                  list2.add("a");              } else {                  list1.add("b");                  list2.add("b");              }          }          test(list1);          test(list2);      }      private static void test(List<String> list1) {          long start = System.currentTimeMillis();          for (int i = 0; i < list1.size(); i++) {              list1.remove("b");          }          long end = System.currentTimeMillis();          System.out.println(end - start);      }  } |

这段代码是删除的 十万条数据

结果两个数组都是前10个位a, 后面的都是b

然后我把b全都删除了

ArrayList用的时间是9秒

LinkedList用的时间是0.07秒

机器不同可能数值不同

但是倍数相差了130多倍, 也很具有说服力了

追问：

这个例子很给力，我自己个先给list里面添加10W条数据，然后再给循环给随机位置添加10W条数据，ArrayList平均去到7000，而LinkedList测只测一次的速度已经不敢恭维了，一次就去到99029，难道我这次的测试误区是在math里面？  
代码如下：  
//List l = new ArrayList（）;  
List l = new LinkedList();  
int q=100000;  
//time 1  
for(int y=0;y<q;1++)  
l.add(i);  
}  
for(int y=0;y<q;1++)  
l.add(math.random\*q,i);  
}//time2

追答：

private static void test(List<String> list) {  
 long start = System.currentTimeMillis();  
 for (int i = 0; i < 100000; i++) {  
 list.add((int)(Math.random() \* 100 + 90000), "c");  
 }  
 long end = System.currentTimeMillis();  
 System.out.println(end - start);  
 }  
你看这段代码, 已有100000条数据,  
  
然后总是在90000 - 90100 这些位置里添元素, 添100000次,  
  
ArrayList : 1554  
LinkedList : 12717  
  
如果是 list.add((int)(Math.random() \* 100), "c");  
总在前100个位置里添元素, 添100000次  
ArrayList : 4514  
  
LinkedList : 19  
  
LinkedList找90000位置还是要话非常多时间的, 甚至比ArrayList添加移动数据话费的时间多很多