## Java 存储结构的查找速度比较

java中有多种存储结构，它们的查找效率如何呢，下面是对他们的比较：

运行代码如下:

**package com.po;**

**import java.util.\*;**

**import java.util.Random;**

**public class Casual**

**{**

**public static void main(String[] args)**

**{**

**ArrayList list=new ArrayList();**

**int i=0;**

**int j=0;**

**final int num=10000000;//查找范围**

**final int sum=20000; //查找的数的数量**

**while(j<num)//顺序存储到ArrayList**

**{**

**int in=++i;**

**list.add(in);**

**j++;**

**}**

**Random rand=new Random();//无序在ArrayList中查找i**

**//System.out.println(it.next());**

**int temp[]= new int[sum];**

**for(int t=0;t<sum;t++)//设置十个随机数来查找**

**{**

**temp[t]=rand.nextInt(num);**

**}**

**long start=System.currentTimeMillis();//求出运行时间**

**Iterator<Integer> it=list.listIterator(0);//从o开始顺序查找**

**i=0;**

**while(it.hasNext())//**

**{**

**if(i==sum)**

**{**

**System.out.println("Have Found "+sum+ " numbers!");**

**break;**

**}//if**

**if(it.next()==temp[i])**

**{**

**i++;**

**it=list.listIterator(0);//从o开始顺序查找**

**}//if**

**}//while**

**long end=System.currentTimeMillis();**

**System.out.println("运行时间是"+(end-start)+"ms!");//输出运行时间**

**//---------------hashset-----------------------**

**System.out.println("\n\nhashSet");**

**Set hs=new HashSet();**

**//temp=10000000;**

**i=0;**

**j=0;**

**while(j<num)**

**{**

**int in=++i;**

**hs.add(in);**

**j++;**

**}**

**i=0;**

**//--------------------------开始查找十个数**

**long Start=System.currentTimeMillis();//求出运行时间**

**while(i!=sum)**

**{**

**if(hs.contains(temp[i]))**

**{**

**++i;**

**if(i==sum)**

**{**

**System.out.println("Have Found "+sum+ " numbers!");**

**}//if**

**}//if**

**}//while**

**long End=System.currentTimeMillis();**

**System.out.println("运行时间是"+(End-Start)+"ms!");//输出运行时间**

**//---------------treeSet-----------------------**

**System.out.println("\n\ntreeSet");**

**Set ts=new TreeSet();**

**i=0;**

**j=0;**

**while(j<num)**

**{**

**int in=++i;**

**ts.add(in);**

**j++;**

**}**

**i=0;**

**//----------------------------------------------**

**long STart=System.currentTimeMillis();//求出运行时间**

**while(i!=sum)**

**{**

**if(hs.contains(temp[i]))**

**{**

**++i;**

**if(i==sum)**

**{**

**System.out.println("Have Found "+sum+ " numbers!");**

**}//if**

**}//if**

**}//while**

**long ENd=System.currentTimeMillis();**

**System.out.println("运行时间是"+(ENd-STart)+"ms!");//输出运行时间**

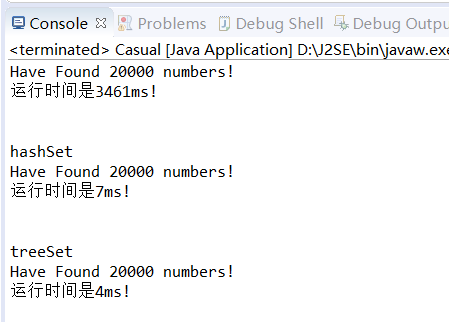
**}//main**

**}//class**

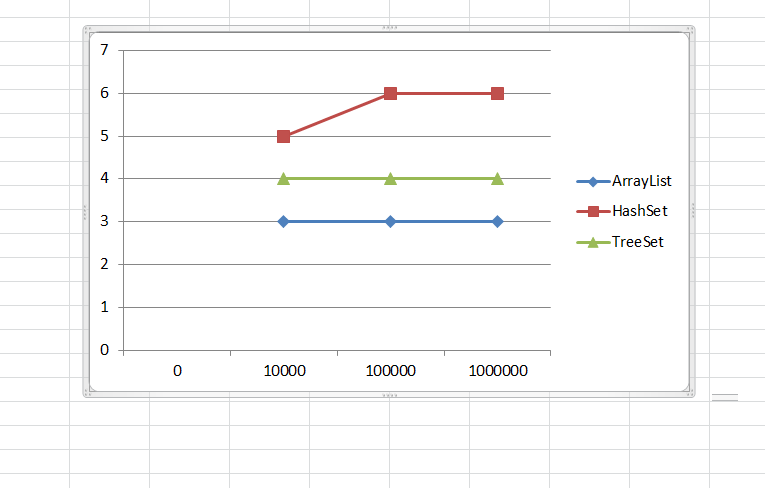
**//Random rand=new Random(1000);**

**//i=rand.nextInt()+1;**

运行截图：



绘制的折线图：



对代码和图的解释：

代码主要是为了比较ArrayList、HashSet、TreeSet的查找效率。利用循环向存储结构中插入多个数再进行查找。其中hashset,treeset采用的是方法object.contains()。从n个数中找2000个数。

图片中的折线图：

横坐标为查找数目的范围，即一万个，十万个，一百万个。

纵坐标为查找2000个数所需的时间，为了让数据在一个表格，ArrayList单位为千微秒，其他的均为微秒。从图中可以直观看出TreeSet查找效率最高。