目录

2023 WustJavaClub 2th 讲义

任务

- 课堂练习：欧拉计划（二）
- 目的：进一步 熟悉 Java 语法（数组、类与对象、集合和泛型）

U265405 013. 大数之和

原题评测

U265405 013. 大数之和 - 洛谷

思路

用 `BigInteger` 存储大数，读取每行求和，然后将该和转化为 `字符串` 并取前十位.

题解

```java
import java.math.BigInteger;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        scan.nextLine(); // 跳过换行符
        BigInteger ans = new BigInteger(scan.nextLine()); // 用 BigInteger 存储大数
        for (int i = 1; i < n; ++i) {
            ans = ans.add(new BigInteger(scan.nextLine())); // BigInteger.add() 是加法
        }
        System.out.println(ans.toString().substring(0, 10)); // 结果转换成字符串取前10位数
    }
}
```

import java.math.BigInteger;

```
BigInteger a = new BigInteger("12345678901234567890");  <-- 注：大数据要用字符串处理的
BigInteger b = new BigInteger("1234567890");
BigInteger sum = a.add(b);    // 加减乘除分别为：add、subtract、multiply、divide
```

6. 获取子串

- substring(int start)：从指定位置开始一直截取到末尾
- substring(int start, int end)：截取 [start,end-1] 范围子串

```
String s1="time is over";
String s2=s.substring(2)        // s2="me is over"
String s3=s.substring(1, 3)     // s3="im"
```

-

-

U265181 016. 指数各位数之和

原题测评

U265181 016. 指数各位数之和 - 洛谷

思路

BigInteger 的 移位操作，求出 2^n，将其转化字符串，并将 字符串各位转化为整数 再求和.

题解

```java
import java.math.BigInteger;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt(), sum = 0;
        String num = BigInteger.ONE.shiftLeft(n).toString(); // shiftLeft(n) 向左移位 n 位，每移1位代表当前值*2
        for (int i = 0; i < num.length(); i++) {
            sum += num.charAt(i) - '0'; // 字符串遍历每一位转换成整型值求和
        }
        System.out.println(sum);
    }
}
```

2. 字符串长度和字符获取

- length()：获取串的长度（字符个数）
- charAt(int index)：获取指定索引位置处的字符(char) ← index索引值从0开始

```
String s="student";
s.length()          //7
s.charAt(2)         // 'u'  ——→ 字符串不能用下标形式：s[2] ×
"中国".length()     //2
"ABC\\n\\u4e2d\u6587".length()    //12  想一想
```

```
String s="time over";
for( int i = s.length() - 1; i >=0; i-- ) {
    System.out.print( s.charAt(i) );  ——→ 逆序输出串
}
```

> 注：【 BigInteger.shiftLeft() 方法详解】->Java.math.BigInteger.shiftLeft()方法实例 - Java.math包 (yiibai.com)

-

-

U273774 022. 姓名分值

原题测评

U273774 022. 姓名分值 - 洛谷

思路

读入英文名字后，先将英文名字的双引号（去除），然后通过逗号（分隔）成字符串数组，升序（排序）后计算各字母分值并求和，最后计算姓名得分.

题解

```java
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String line;
        StringBuilder str = new StringBuilder(); // 需要修改 str 用 StringBuilder
        /* 因输入不定行，输入完后需要 Ctrl+D 手动结束输入 */
        while (scan.hasNext()) {
            line = scan.nextLine();
            line = line.replace("\"", ""); // 将双引号去除
            str.append(line);
        }
```

```
        String[] names = str.toString().split(","); // 以逗号分割
        Arrays.sort(names); // 数组默认升序排序
        long ans = 0;
        for (int i = 0; i < names.length; ++i) {
            long sum = 0;
            for (int j = 0; j < names[i].length(); ++j) {
                sum += names[i].charAt(j) - 'A' + 1; // 字符串遍历每一位转换成分值求和
            }
            ans += sum * (i + 1);
        }
        System.out.println(ans);
    }
}
```

9. 字符串替换

注：替换不会改变原字符串的内容，而是返回了一个新字符串

- replace(String oldS, String newS)：用新串newS替换老串oldS

```
String s1 = "time over";
String s2 = s1.replace("e","yy");    // s2="timyy ovyyr"，s1不变
```

12. 关于 String、StringBuffer、StringBuilder

- String字符串是不可改变的，也就是说一旦创建了String对象，那它的值就无法改变！
- 要对字符串自身进行修改，可使用StringBuffer类
- StringBuffer对象能被多次修改，且不产生新的对象！
- StringBuilder和StringBuffer本质上没区别，只是StringBuilder去掉了线程安全，不同步，效率更高。

【附录2】Arrays类用法

import java.util.Arrays;

- Arrays是针对数组的工具类，提供了排序，查找，复制填充，二分查找等功能， 大大提高了开发效率

Arrays常用方法	说明
sort(array)	对指定的基本数据类型数组array按升序排列
binarySearch(array,val)	对基本数据类型数组array进行二分查找val
equals(array1,array2)	如果两个指定的基本数据类型数组长度和内容都相等返回true
fill(array,val)	将指定的基本数据类型值数组array的所有元素都赋值为val (填充)
copyof(array,length)	把基本数据类型数组array复制成一个长度为length的新数组
toString(array)	把基本数据类型数组array内容转换为字符串
asList(T... a)	把数组a转换成List集合（注：不可以使用集合的增删方法）

-

-

U274266 026. 倒数周期

原题测评

U274266 026. 倒数周期 - 洛谷

思路

要求循环节的长度，我们可以 建立一个列表记下每次除法中的被除数，如果被除数小于除数，则需要将被除数进一位再去除，所得余数为下一次除法中的被除数，如此循环下去，知道当前被除数在列表中已经出现过了，之后的计算会重复之前的模式，此时一个循环已经完成，列表的长度就是循环节的长度.

因此，不难推出 余数（下次被除数）的计算公式:

$$余数 = 余数 * 10 \ mod \ n \tag{2}$$

例如，对于 $1/7$ 计算循环节长度的过程如下:

$$\begin{aligned}
1 \ mod \ 7 &\rightarrow [1] \\
10 \ mod \ 7 = 3 &\rightarrow [1,3] \\
30 \ mod \ 7 = 2 &\rightarrow [1,3,2] \\
20 \ mod \ 7 = 6 &\rightarrow [1,3,2,6] \\
60 \ mod \ 7 = 4 &\rightarrow [1,3,2,6,4] \\
40 \ mod \ 7 = 5 &\rightarrow [1,3,2,6,4,5] \\
50 \ mod \ 7 = 1 &\rightarrow [1,3,2,6,4,5,1]
\end{aligned}$$

同时，为了 判断当前被除数在列表中是否已经出现过，可以用`HashMap.containsKey()`方法，因为 HashMap 是保证主键唯一 的（ Map 的 主键存放余数，值为循环执行的次数）.

题解

```java
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Main {
    /**
     * 计算 1/d 的循环节长度
     *
     * @param d 单位分数的分母
     * @return 1/d 的循环节长度
     */
    public static int getLoopLength(int d) {
        Map<Integer, Integer> map = new HashMap<>(); // map（泛型 Map）的主键存放余数，值为循环执行的次数
        int remainder = 1; // 第一个余数为1
```

```
        for (int cycleNumber = 0; ; ++cycleNumber) {
            if (map.containsKey(remainder)) {
                return cycleNumber - map.get(remainder); // 循环次数差才是循环节长度
            } else {
                map.put(remainder, cycleNumber);
                remainder = remainder * 10 % d; // 求下次被除数
                if (remainder == 0) return 0; // 如果余数为0，则表示能够除尽，循环节长度为0
            }
        }
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt(), ansNum = 1, ansLoopLength = 0;
        for (int i = 1; i < n; ++i) {
            int loopLength = getLoopLength(i);
            if (loopLength > ansLoopLength) {
                ansLoopLength = loopLength;
                ansNum = i;
            }
        }
        System.out.println(ansNum);
    }
}
```

Map实现类：

接口	简述	实现类	操作特性	成员要求
Map	键值对成员基于键找值操作	HashMap 常用	底层采用哈希表结构，元素的存取顺序不能保证一致。由于要保证键的唯一、不重复，需要重写键的hashCode()方法、equals()方法。	键成员可为任意Object对象
		TreeMap 有序的	支持对键有序地遍历，附加实现了SortedMap接口，支持要求顺序的操作	键成员要求实现Comparable或使用Comparator接口，TreeMap键成员一般为同一类型。
		LinkedHashMap 保证键值对顺序	底层采用的哈希表+双向链表结构。通过链表结构保证键值对的插入顺序；通过哈希表结构可以保证的键的唯一、不重复，需要重写键的hashCode()方法、equals()方法。	成员与HashMap成员类似

Map基本操作

- 查询操作：
  - V get(Object key)：根据key取得对应的值（通常需要强转）
  - boolean containsKey(Object key) ：判断Map中是否存在某key
  - boolean containsValue(Object value)：判断Map中是否存在某值value
  - int size()：返回Map中键值对的个数
  - boolean isEmpty()：判断当前Map是否为空

泛型Map示例1：

假设key和value都是String类型

```
Map<String, String> map = new HashMap<String, String>();
map.put("武汉科技大学", "https://www.wust.edu.cn");
map.put("华中科技大学", "https://www.hust.edu.cn");
for ( Map.Entry<String, String> entry : map.entrySet() ) {
    String mapKey = entry.getKey();
    String mapValue = entry.getValue();
    System.out.println(mapKey + "： " + mapValue);
}
```

使用for-each循环遍历entrySet()

-

-

U275835 029. 独特的幂

原题评测

U275835 029. 独特的幂 - 洛谷

思路

暴力遍历，使用 `BigInteger` 计算幂结果，并使用 `泛型 Set` 存储计算结果，用 `HashSet去重` .

题解

```java
import java.math.BigInteger;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int left1, right1, left2, right2;
        left1 = scan.nextInt();
        right1 = scan.nextInt();
        left2 = scan.nextInt();
        right2 = scan.nextInt();
        Set<BigInteger> set = new HashSet<>(); // 泛型集合，HashSet 去重
        for (int i = left1; i ⩽ right1; ++i) {
            for (int j = left2; j ⩽ right2; ++j) {
                set.add(BigInteger.valueOf(i).pow(j)); // x.pow(y) 求 x 的 y 次方
            }
        }
        System.out.println(set.size());
    }
```

```
22 }
```

Set实现类

接口	简述	实现类	操作特性	成员要求
Set	成员不能重复	HashSet 无序，唯一	外部无序地遍历成员 (查找效率高)	成员可为任意Object子类的对象，但如果覆盖了equals方法，同时注意修改hashCode方法。
		TreeSet 有序，唯一 底层是自平衡二叉树结构(红黑树)	外部有序地遍历成员； **特殊：**附加实现了SortedSet, 支持子集等要求顺序的操作	成员要求实现Comparable接口，或者使用Comparator构造TreeSet。成员一般为同一类型(不能为null)。
		LinkedHashSet	外部按成员插入顺序遍历成员	成员与HashSet成员类似