# 算法刷题

* 输入的数字是不是回文数
* 描述：判断一个数是否是回文数(回文字符串)
* 回文数也是一个数字，数字的特点是正反序是同一个数字。
* 思路如下：

|  |
| --- |
| * 思路一： * 控制台输入数字，将数字转换成字符数组，对数组进行遍历，通过boolean变量记录是否是回文数 |

* 具体代码如下：

Scanner scan = new Scanner(System.*in*);  
BigInteger temp = new BigInteger(String.*valueOf*(scan.nextBigInteger()));  
try {  
 //将数字放进字符数组中

char[] tempArr = temp.toCharArray();

boolean isPalindrome = false;  
 flag:for(int i=0;i<tempArr.length/2;i++){  
 for(int j=tempArr.length-1-i; j>tempArr.length/2;j--){  
 if(tempArr[i]== tempArr[j]){  
 isPalindrome = true;  
 break;  
 }else{  
 isPalindrome = false;  
 break flag;  
 }  
 }  
 }  
 if (isPalindrome){  
 System.*out*.println(temp+"是回文");  
 }else{  
 System.*out*.println(temp+"不是回文");  
 }  
} catch (Exception e) {  
 e.printStackTrace();  
 System.*out*.println("只允许输入数字");  
}

* 用了两个for循环，效率不高

|  |
| --- |
| Scanner scan = new Scanner(System.*in*); BigInteger temp = new BigInteger(String.*valueOf*(scan.nextBigInteger())); try {  //将数字放进字符数组中  char[] tempArr = temp.toCharArray();   boolean isPalindrome = false;  for (int i=0;i<tempArr.length/2;i++){  if(tempArr[i]!=tempArr[temp.length()-1-i]){  isPalindrome = false;  break;  }else{  isPalindrome = true;  continue;  }  }  System.*out*.println("是否是回文数："+isPalindrome); } catch (Exception e) {  e.printStackTrace();  System.*out*.println("只允许输入数字"); } |

* 用一个for循环，一个从前往后，一个从后往前，长度控制在数字长度的1/2
* 时间复杂度：
* 的