LeetCode刷题笔记

快慢指针可用于链表中诸如环的处理。

快慢指针也可以用于在链表中获得中间节点，然后将链表对半分！

占字节多的数据类型强制转为占字节少的数据类型时是截断前面的位置，保留后面的数据

注意正数的原码、反码、补码一致，负数的反码是原码除符号位外取反，补码是反码加1.计算机中以补码存数，最高位表示符号。

byte占一个字节，取值为-128（补码为10000000）到127（01111111）

Java中的byte、short、char在计算时会隐形的提升为int型计算，所以把2个byte计算结果赋值给一个byte是错误的

如果byte、short、char有final修饰则不会隐式类型转换

Java中采用unicode编码，中英文字符都是占两个字节

注意递归的边界问题！注意测试样例可能有很多种，不能少了一些奇葩的测试样例

从旋转数组中发现了三步翻转法！

从顺时针旋转矩阵中发现规律，顺时针旋转n\*n的矩阵90度相当于沿着副对角线调换元素再调换每行元素（如第0行和第n-1行换，第1行和第n-2行换）

Charcter的静态方法getNumericValue可以将'9'字符返回数字9.

光标移动到某个类上按f3可以查看这个类的源代码

java中字符串可以用+连接形成新的字符串

可以用两个队列来实现对树的非空节点进行BFS遍历其值的操作

对java几种数据结构的理解：

1.数组：相对比较简单，注意下标不要超过限制，尤其写判断循环条件的时候看下会不会引用到OutOfBound的元素

2.链表：注意使用链表遍历或者快慢指针的方法得到我们想要的某个节点

3.树：非常喜欢的数据结构，注意经常可以使用递归结构（如可以用于实现DFS），以及用队列可以实现BFS遍历

4.堆：本质上是数组形式的结构，每个节点有index，这是和树不一样的结构，但是逻辑和树比较像

x>>>=1表示x二进制无符号右移一位后的值，无符号右移指左边补0

>>和<<分别表示右移和左移

x&1表示位与操作取出x二进制表示的最后一位

遍历比递归快，但是解决的时间在一个数量级上，不过递归要注意栈溢出的问题，尤其是杨辉三角那题

计数素数要用筛法，暴力解法太慢了

ArrayList<Integer>arrayList=new ArrayList<>();

System.out.println(arrayList.size());会显示0

System.out.println(arrayList.get(0));会数组下标越界报错