## 异或性质：

补充，异或运算的简单性质：

　1. a ⊕ a = 0

　2. a ⊕ b = b ⊕ a　　// 异或运算满足交换律

　3. a ⊕ b ⊕ c = a ⊕ (b ⊕ c) = (a ⊕ b) ⊕ c; 异或运算满足结合律

　4. d = a ⊕ b ⊕ c 可以推出 a = d ⊕ b ⊕ c.

　5. a ⊕ b ⊕ a = b.

　6.若x是二进制数0101，y是二进制数1011

7 a ⊕0=a

　则x⊕y=1110

　只有在两个比较的位不同时其结果是1，否则结果为0

　即“相同为0，不同为1”.  
　由上述这些性质我们可以得到异或运算的第二个应用————判断两个数是 否相等。

## 特殊用法总结：

* 是不是偶数：a&1 结果得1表示奇数，得0表示偶数
* a>>1 除以2; a<<1乘以二, a<<i表示a乘2的i次方
* lowbit(a)=a&(-a)表示二进制下第一个1的位置，树状数组用
* a^1 表示若a是奇数就减一，是偶数就加一
* 取二进制下a的第i位数字：(a>>(i-1))&1 第几位从1开始
* Void swap(int &a,int &b){ a ^= b^= a^= b; }