**当一个无符号整数x是非2的幂次方时候怎么求大于x且最接近x的2的幂次**

**的数**

**static** **public** **int** nextPowerOfTwo (**int** value) {

**if** (value == 0) **return** 1;

value--;

value |= value >> 1;

value |= value >> 2;

value |= value >> 4;

value |= value >> 8;

value |= value >> 16;

**return** value + 1;

}

原理：将该数的最高位为1的二进制后面的数都变为1，最后再加上1即可

value >> 1保证最高位为1相邻的位变为1

value >> 2保证最高位为1的相邻3为变为1，依次类推

**判断点在多边形内**

1)面积法（凸多边形）

运用海伦公式S = sqrt(p\*(p-a)\*(p-b)\*(p-c)) p = (a+b+c)/2

2)夹角和判别法

判断目标点与所有变的夹角和是否为360度

3）射线法

<http://blog.csdn.net/hjh2005/article/details/9246967>

其核心思想是由点p向x正方向发射一个射线，穿过多边形线段上的个数为奇数则在多边形内，偶数则在多边形外

libgdx的实现：

**public** **boolean** contains (**float** x, **float** y) {

**final** **float**[] vertices = getTransformedVertices();

**final** **int** numFloats = vertices.length;

**int** intersects = 0;

**for** (**int** i = 0; i < numFloats; i += 2) {

**float** x1 = vertices[i];

**float** y1 = vertices[i + 1];

**float** x2 = vertices[(i + 2) % numFloats];

**float** y2 = vertices[(i + 3) % numFloats];

**if** (((y1 <= y && y < y2) || (y2 <= y && y < y1)) && x < ((x2 - x1) / (y2 - y1) \* (y - y1) + x1)) intersects++;

}

**return** (intersects & 1) == 1;

}

**如何判断一个单向链表有环？**

**Kmp算法**

**红黑树**

**非递归遍历二叉树**

**排序（基数排序、shell排序、桶排序、其他排序及非递归方式）**

**八数码问题**

**Kd-tree**

**R-tree**

**A\*算法**

**并查集**

**伙伴系统**

**牛顿迭代**

**字符串全排列**

**Stl**