**思想作者：罗瑶光**

**功能作者：罗瑶光**

**1快速排序的逻辑 略**

2 快速排序的递归是基于2分拆分数组进行递归迭代的

2.1 假设数组的长为n， 那么 2分拆分将会有2种结果。

2.2 当n 为偶数时，则为完整拆分。

2.3 当n 为基数时，问题来了。

2.3.1 n/2 将会有一定的概率 在2分递归的时候不对等。这个不对等产生的分配概率会形成许多短暂的运算小高峰时序。

概率的拆分如图：

基数产生缺陷

如果中间拆分时为基数，将会产生概率分配误差的噪声峰，基于左右对比来确定一个线性方向，可以快速降解这个峰值。

另外当一个单元的while循环次数增多时，峰值的利用增大，有效的计算性能增加。

于是确定线性方向x 选边缘最大数值。

最后采用迪摩根分配律进行微分条件，性能再加速。

于是比对算法快速排序有效的平滑这个问题：如下：

**private** **int** partition(**int**[] a, **int** lp, **int** rp) {

**int** x= a[lp]>= a[rp]? a[lp]: a[rp];

**int** lp1= lp;

**while**(lp1< rp){

**while**(!(a[lp1]>x|| lp1>=rp)) {

lp1++;

}

**while**(a[rp]>x){

rp--;

}

**if**(lp1<rp){

**int** temp=a[rp];a[rp]=a[lp1];a[lp1]=temp;

}

}

a[lp]=a[rp];a[rp]=x;

**return** rp;

}