**设*X*[0:n-1]和*Y*[0:n-1]为两个数组，每个数组中含有*n*个已排好序的数。试设计一个O(log*n*)时间的分治算法，找出*X*和*Y*的2*n*个数的中位数，并证明算法的时间复杂性为O(log*n*)**

* + 个数为奇数,则处于最中间位置的数
  + 个数为偶数,则中间两个数据的平均数

算法分析：

两个数组，X和Y。设X数组的第一个、中间、最后一个元素分别为lx，mx，rx；设X数组的第一个、中间、最后一个元素分别为ly，my，ry。

先分别求两个数组的中位数mx,my进行比较，如果mx < my，比如X的中间数是3，Y的是5，那么推测两个数组的中间数约等于4，则该中间数对于X来说一定在3的右边，即在mx到rx范围的子数组里面，对于Y来说，中间数一定在5的左边，即在ly到my的子数组里面。比较如下：

if mx < my:

新mx属于[mx, rx]

新my属于[ly, my]

else:

新mx属于[lx, mx]

新my属于[my, ry]

之后再对新的两个子数组进行求中位数比较，一直到子数组元素为1时，进行比较，即可找到原数组X和Y的中位数。

新的mx和my将原来的数组X和Y分别分成了两个子数组，但是在四个子数组中只需使用上式指定的两个子数组即可，所以一次分解之后为原问题规模更小的实例T(n/2)，计算复杂度仍为计算两个数组的中位数（和原问题一样），为T(n/2)。求中间数和中间数比较需要的时间复杂度为O(1),故总的递归式为：

T(n) = T(n/2) + O(1)

计算可得时间复杂度为T(n) = O(logn)