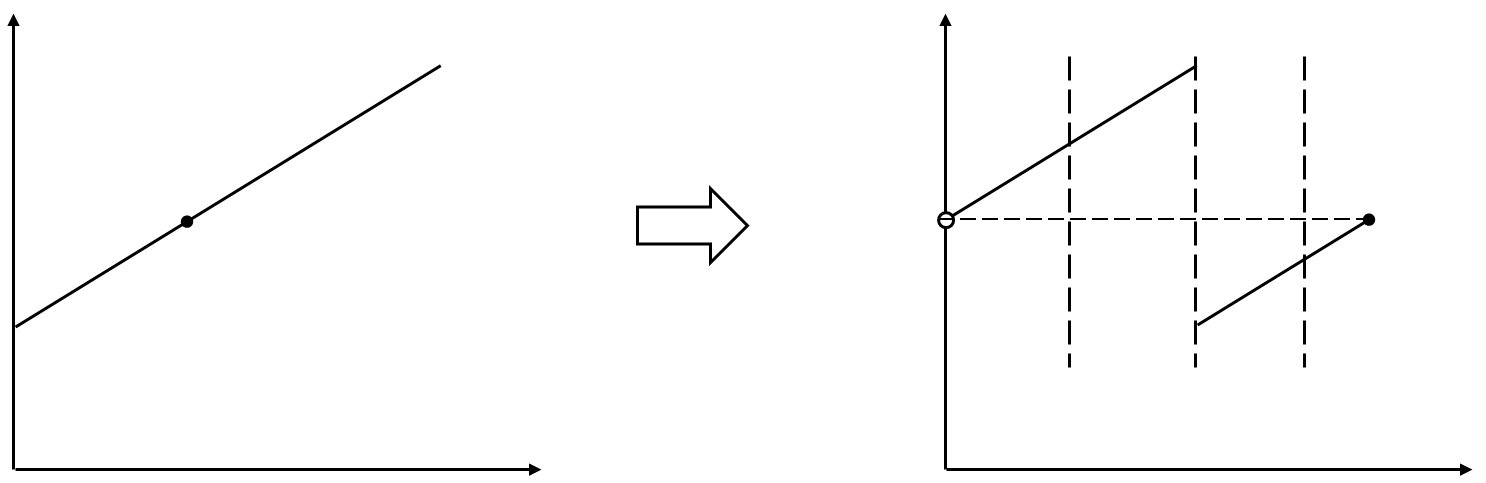
1. 课本172页**解决的问题1**：

对于一个形如山峰的数组，找到它的峰值。

如上图所示，横轴为数组元素的下标，纵轴为元素大小，可以看到，元素先由小变大，之后再由大变小，我们的目标就是找到数组的峰值。

分析：如果使用遍历的方法，则O(n)就可以求出峰值，即最大值。但如果使用二分查找的思想，我们取当前数组中间的元素，并进行划分，然后判断轴元素与相邻两个元素的大小关系，如果x1<p<x2，则说明最大值在右侧，我们对右子数组继续进行求解。同理，如果x1>p>x2，则对左子数组进行求解，如果p>x1且p>x2，则p就是我们要找的元素。该算法的复杂度仅为O(logn)。

变式：对于一个严格递增的序列，A1, A2, …, An，我们选择一个位置k进行旋转，从而生成一个新的序列Ak+1, Ak+2, …, An, A1, A2, …, Ak。例如一个序列为{1, 2, 3, 4, 5, 6, 7}，选择k=2进行旋转，生成的新序列为{3, 4, 5, 6, 7, 1, 2}，注意Ak并不是轴。对于一个新序列，请设计一算法，求解该序列的最小值。



分析：同样使用二分查找的方法，在数组中间取一个元素x，并与数组最后一个元素r进行比较，如果x>r，则k点在x右侧，如果x<r，则x在r左侧。当然，首先要判断一下，x的右侧一个元素是否比x小，如果小，则该右侧的值就是序列最小值。当然，如果要求解第k小的元素，则从最小值往右数k个，如果大于n则-n即可得到序列第k小的值。

注意：本题要求严格递增，即不仅要有唯一的最小值，而且元素各不相同。

1. 常见分治问题的复杂度：
2. **T(n) = T(n/2) + c = O(logn)**

**a = 1, b = 2, k = 0, p = 0, case 2**

将问题分成两个子问题，并且能够在常数时间内确定解所在的子问题区间，并丢弃另外一个子问题。如二分查找、上面的确定峰值问题等。

1. **T(n) = 2T(n/2) + c = O(n)**

**a = 2, b = 2, k = 1, case 1**

将问题分成两个子问题，并且能够在常数时间内完成对根节点的操作，然后分别解决两个子问题。如平衡二叉树的遍历（前序、中序、后序类似）问题等。

1. **T(n) = 2T(n/2) + θ(n) = O(n\*log n)**

**a = 2, b = 2, k = 1, p = 0, case 2**

将问题分成两个子问题，并且需要对两个子问题进行O(n)级别的合并或判断跨越两个子问题的解的存在性。如归并排序、逆序数对、随机快排、平面最近点对、股票买卖、连续子序列最大和问题等。

1. **T(n) = T(n/2) + θ(n) = O(n)**

**a = 1, b = 2, k = 0, case 3**

将能够在O(n)时间将问题分成两个子问题，并且能够在常数时间内丢弃一个子问题，选择另一个子问题递归求解。如求解序列中位数或第k大元。当然，也可以是常数时间分割，但在O(n)时间内选择一个子问题。

1. 分治法复杂度主定理应用
2. 试分析 T(n) = 2T(n/2) + θ(n/logn) 的复杂度的上下界

解：上界可以通过将logn去除，得到a=2, b=2, k=1, p=0, T’(n)=n\*logn

下界可以将n替换为logn，得到T’’(n)=n

所以该问题的复杂度介于[nlogn, n]之间，本质上是nlog(log n)的复杂度。

1. T(n) = T(n/2) + T(n/4) + T(n/8) + θ(n)，则T(n)=O(n)

结论，前面T(x)中的x相加如果严格小于θ(x)中的x，则后面的占主要，所以本题复杂度为O(n)

1. 比较T1(n) = T(3/4 n) + c 与T2(n) = 3\*T(n/4) + c的复杂度大小。

解：根据主定理，对T1(n)，有a=1, b=4/3, k = 0, 所以复杂度为logn。

对T2(n)，有a=3, b=4, k = log43 < 1, 所以复杂度为n^(log43)，所以T2(n)> T1(n)，二者相差是指数级别的。

复习：根据换底公式，在渐进意义下，对数形式的复杂度的底数是不重要的，而指数形式的复杂度的底数是重要的（详见课本30页分析）。

1. 比较T1(n) = T(3/4 n) + n 与T2(n) = 3\*T(n/4) + n的复杂度大小。

解：由c)的分析易知，T1(n)=O(n)=T2(n)，因为两者的k都比1小，所以复杂度相同。