# 第6章 《变治法》习题

1. 考虑这样一个问题：它要找出*n*个数字构成的一个数组中两个最接近数的距离（两个数*x*和*y*之间的距离定义为*x*−*y*）。

（1）设计一个基于预排序的算法来解决该问题，并确定其效率类型。

（2）对该算法的效率类型和蛮力法的效率类型进行比较。

（1）：

PresortElementUniqueness(A[0..n-1])

min = MAX\_VALUE

for i ← 0 : n-1

if A[i] – A[i+1] < min

min = A[i] – A[i+1]

end if

end for

return min

O(nlogn) + O(n) = O(nlogn)，即效率类型为O(nlogn)

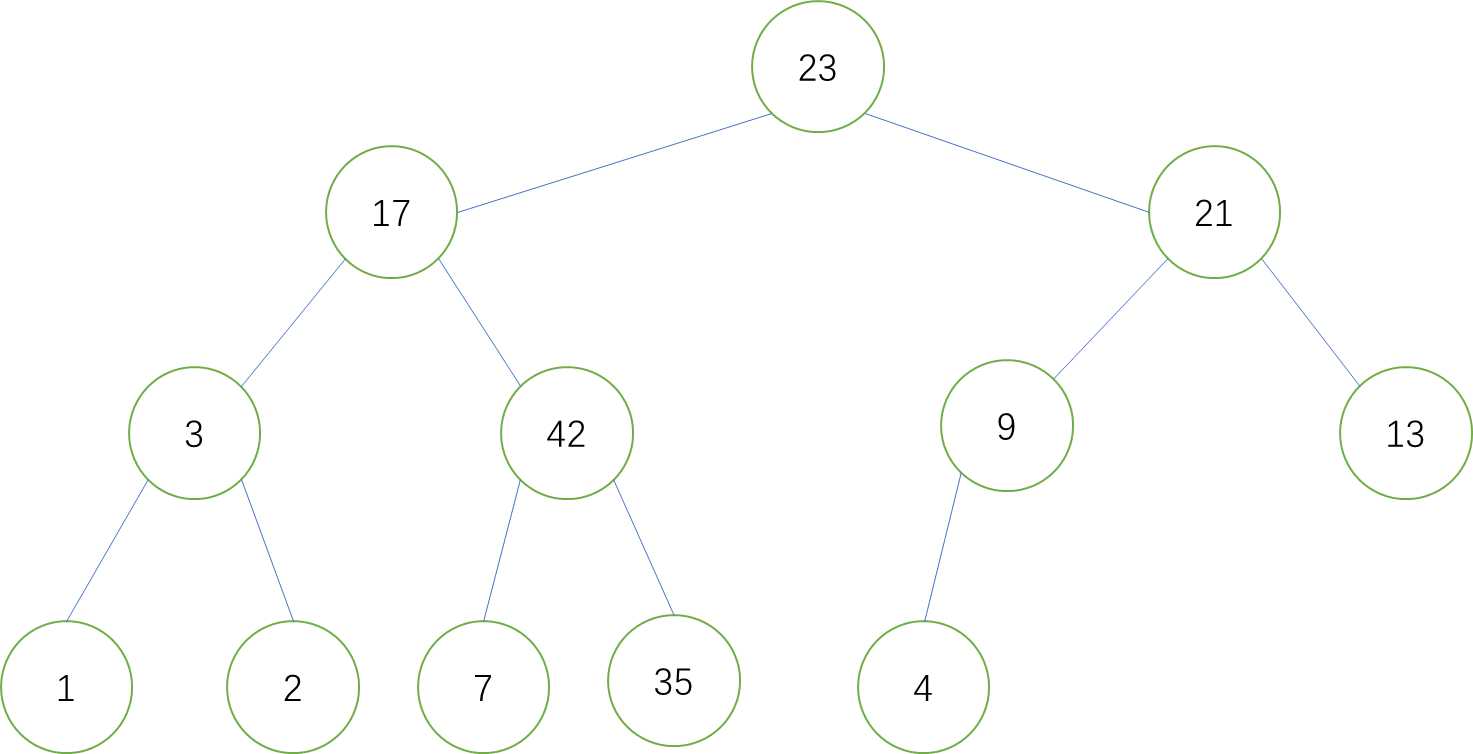
（2）：

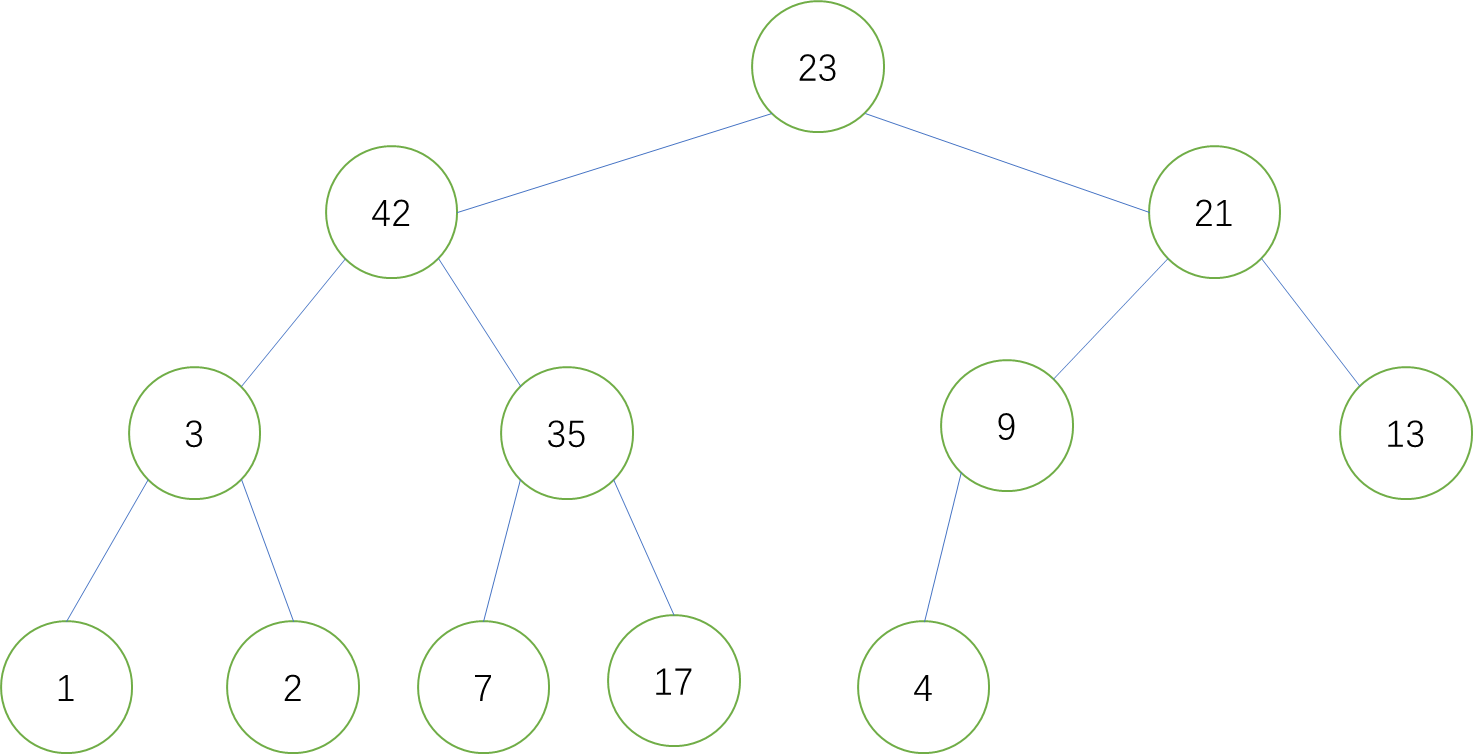
蛮力法需要双重遍历，时间复杂度为O(n2)

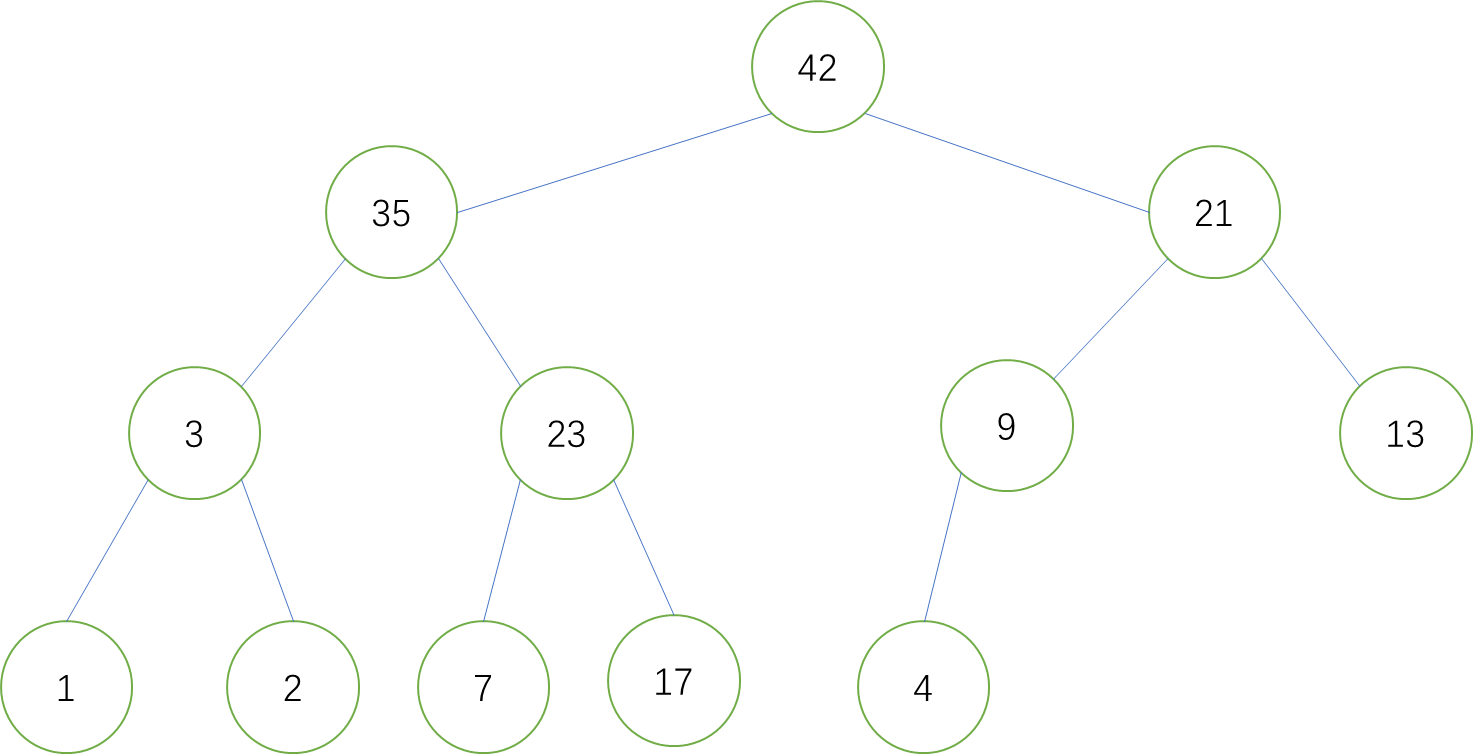
很明显，虽然（1）中算法需要先进行排序，但是时间复杂度仍小于蛮力法

2. 利用图示法给出对以下列表构造最大堆的过程：

[23，17，21，3，42，9，13，1，2，7，35，4]







3. 给定列表*A*[1:19]，为其构造一个堆。

（1）sift-down操作将执行多少次？为什么？

（2）给出一个具有最少元素比较次数的例子，给出其中比较操作执行的次数，并进行解释说明。

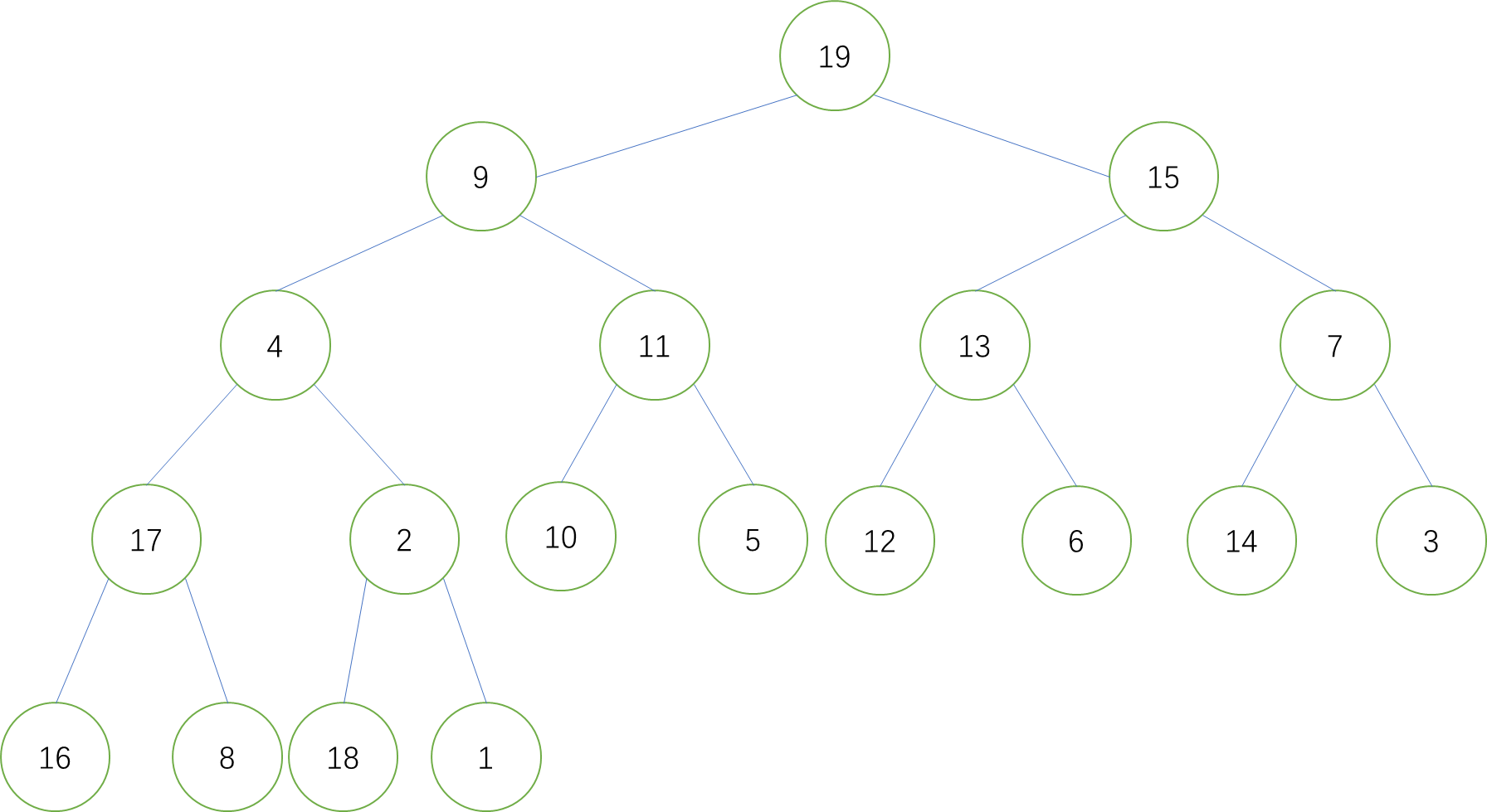
（1）：

16次。

　　因为列表是递增，所以对于A[1:19/2]，中的每个元素，都需要进行sift-down操作，每个元素进行sift-down操作的次数取决于以这个元素为根的树的高度，所以总操作次数为：1\*5 + 2\*2 + 3\*1 + 4\*1 = 16次。

（2）：

　　反向来看，对于一个已经构建好的大堆，在缩小堆排序数组时要想比较次数最多就需要交换次数最多，要交换次数最多，就需在堆顶元素与堆尾元素交换后下沉最多，要想下沉最多可以在堆顶元素与堆尾元素交换时让堆尾元素是堆中的最小元素。按照这个方法就可以构建一个比较次数最多的大堆。反过来操作就可以得到一个比较次数最少的大堆。



共执行sift-down操作6次，故比较次数为12次