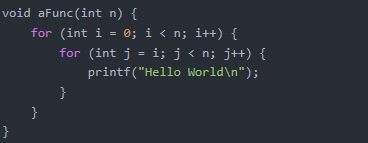
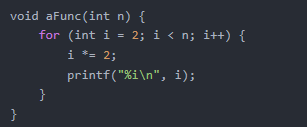
离散数学

时间复杂度计算

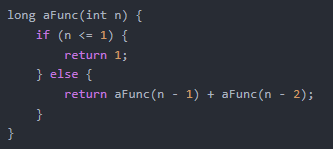


执行次数 T(n) = n + (n - 1) + (n - 2)……+ 1 = n(n + 1) / 2 = n^2 / 2 + n / 2。



则循环条件满足 2^t < n。

可以得出，执行次数t = log(2)(n)，即 T(n) = log(2)(n)，可见时间复杂度为 O(log(2)(n))，即 O(log n)。



2^n

数据结构

顺序表，链表

顺序栈，链栈

循环队列

二叉树

图遍历

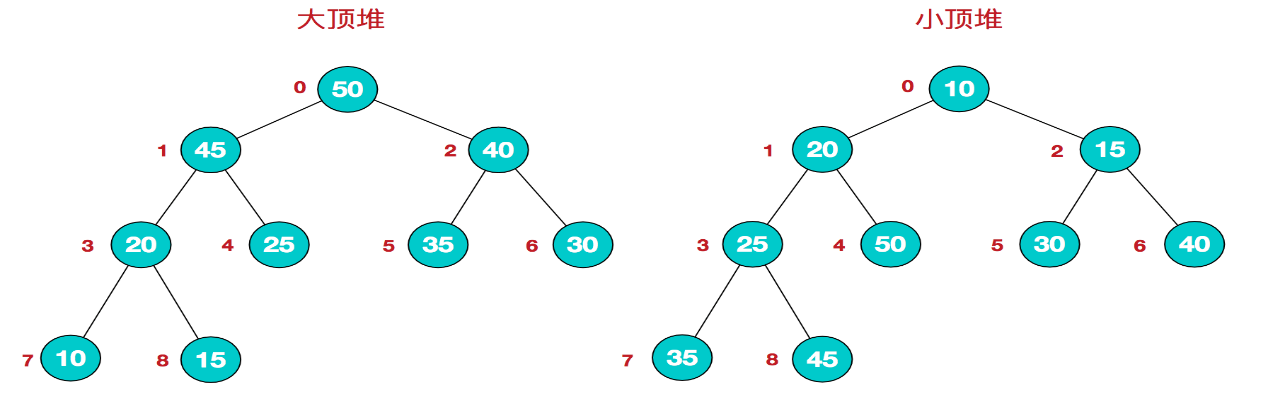
算法

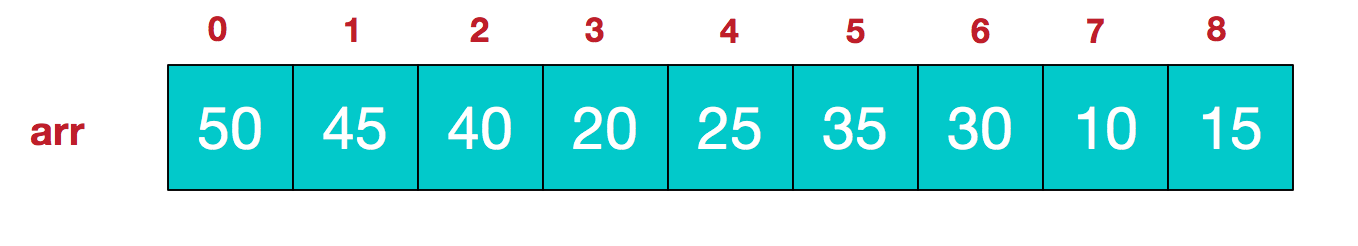
**1、排序**

插入排序，冒泡排序，归并排序，快速排序

**堆排序：**

堆是具有以下性质的完全二叉树：每个结点的值都大于或等于其左右孩子结点的值，称为大顶堆；或者每个结点的值都小于或等于其左右孩子结点的值，称为小顶堆。如下图：



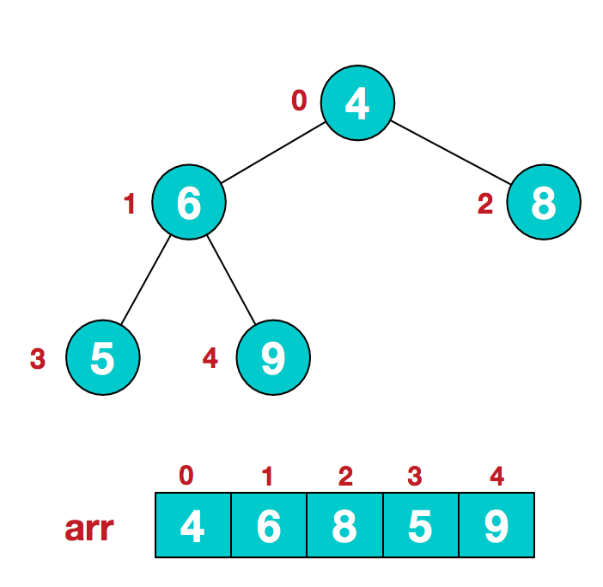


大顶堆：arr[i] >= arr[2i+1] && arr[i] >= arr[2i+2]

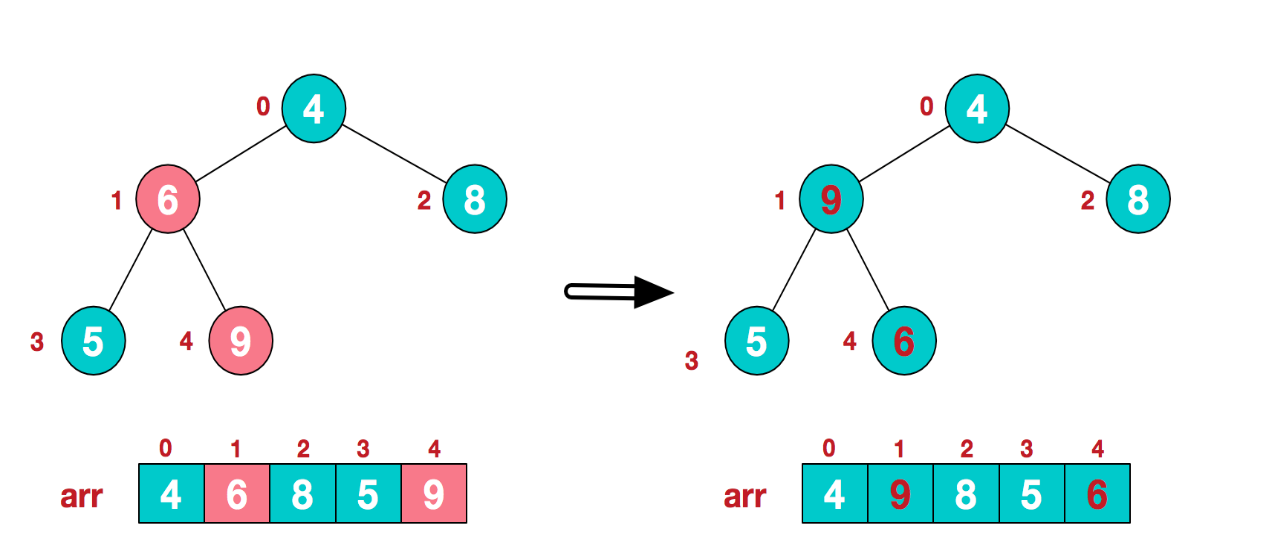
小顶堆：arr[i] <= arr[2i+1] && arr[i] <= arr[2i+2]

堆排序：

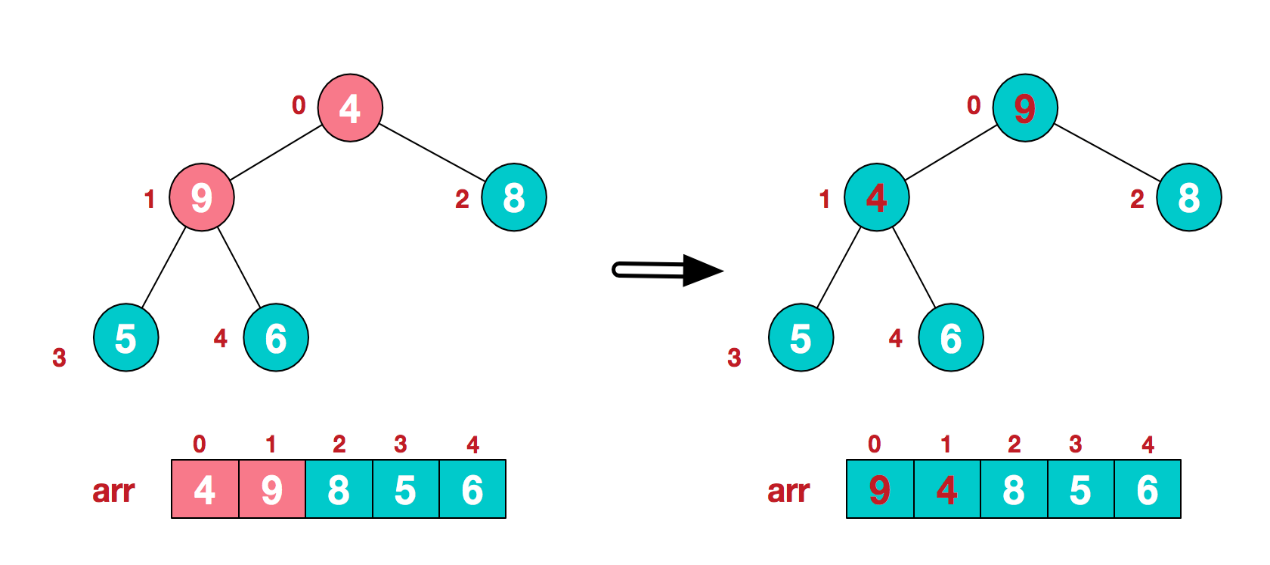
**步骤一 构造初始堆。将给定无序序列构造成一个大顶堆（一般升序采用大顶堆，降序采用小顶堆)。**



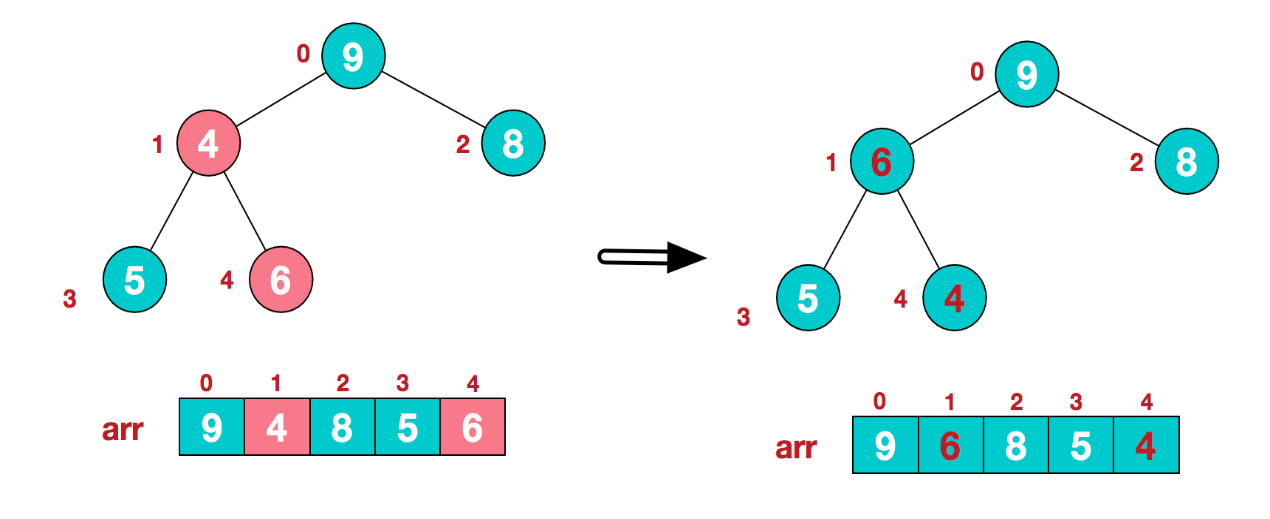
2.此时我们从最后一个非叶子结点开始（叶结点自然不用调整，第一个非叶子结点 arr.length/2-1=5/2-1=1，也就是下面的6结点），从左至右，从下至上进行调整。



找到第二个非叶节点4，由于[4,9,8]中9元素最大，4和9交换

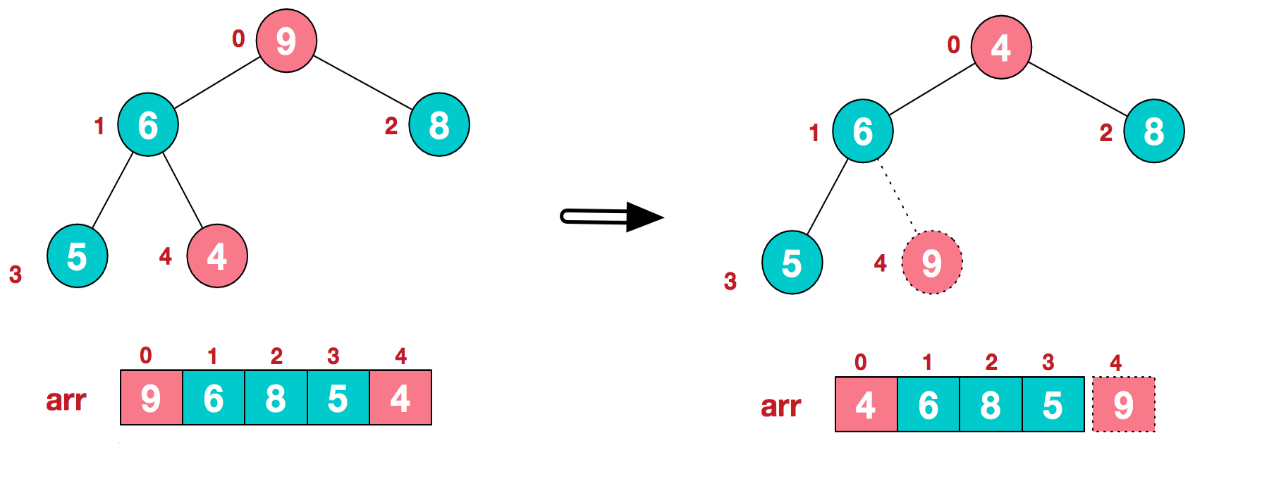


这时，交换导致了子根[4,5,6]结构混乱，继续调整，[4,5,6]中6最大，交换4和6



此时，我们就将一个无需序列构造成了一个大顶堆。

**步骤二** 将堆顶元素与末尾元素进行交换，使末尾元素最大。然后继续调整堆，再将堆顶元素与末尾元素交换，得到第二大元素。如此反复进行交换、重建、交换。

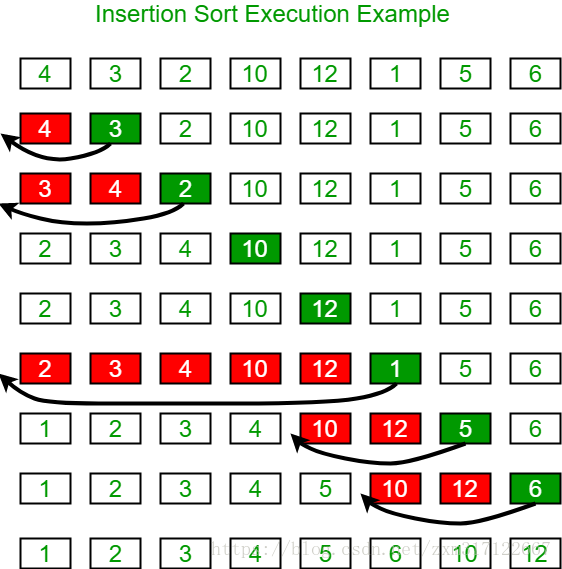


综上：a.将无需序列构建成一个堆，根据升序降序需求选择大顶堆或小顶堆;

　　b.将堆顶元素与末尾元素交换，将最大元素"沉"到数组末端;

　　c.重新调整结构，使其满足堆定义，然后继续交换堆顶元素与当前末尾元素，反复执行调整+交换步骤，直到整个序列有序。

**直接插入排序**



while (j>=0 && arr[j] > key)

{

arr[j+1] = arr[j];

j = j-1;

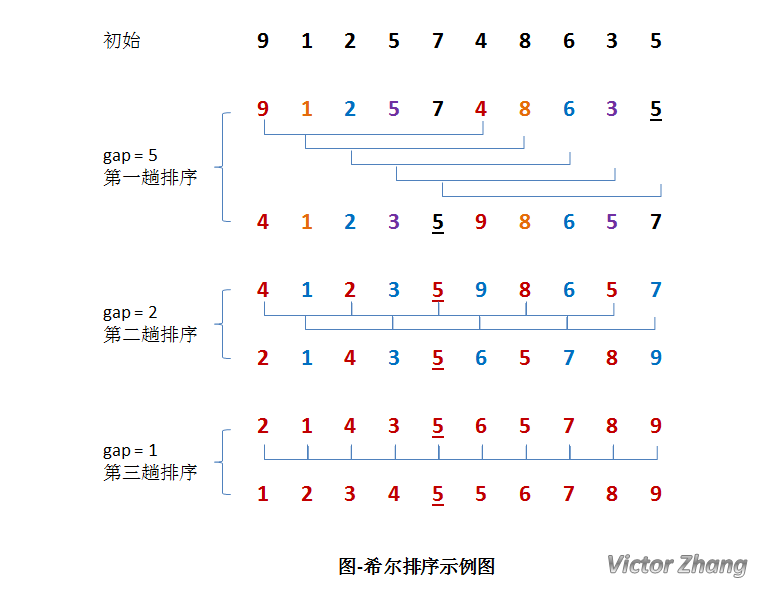
}

arr[j+1] = key;

**希尔排序**

希尔排序（shell sort）这个排序方法又称为缩小增量排序，是1959年D·L·Shell提出来的。该方法的基本思想是：设待排序元素序列有n个元素，首先取一个整数increment（小于n）作为间隔将全部元素分为increment个子序列，所有距离为increment的元素放在同一个子序列中，在每一个子序列中分别实行直接插入排序。然后缩小间隔increment，重复上述子序列划分和排序工作。直到最后取increment=1，将所有元素放在同一个子序列中排序为止。

由于开始时，increment的取值较大，每个子序列中的元素较少，排序速度较快，到排序后期increment取值逐渐变小，子序列中元素个数逐渐增多，但由于前面工作的基础，大多数元素已经基本有序，所以排序速度仍然很快。

[](http://images2015.cnblogs.com/blog/318837/201604/318837-20160422102024757-37862627.png)