**问题1：**

对于一个大小不定的数据流，如何设计一个算法，保留前k大个数据？

**答：**

方法1：

维护一个最小堆，对前k个数据建一个最小堆，之后每来一个数据，将其跟堆顶元素比较，如果更大，则替换，否则舍弃。

方法2：

将数据分成k个一组，对每组进行排序，之后做归并。

两种方法的复杂度均为O(n\*logk)，方法1，建立一个k大小的堆复杂度为O(k)，插入时复杂度为O(logk)，所以对于n个数据，总复杂度为O(k)+O(n\*logk)= O(n\*logk)。

方法2，将数据分成k个一组，然后每组排序，复杂度为O(n/k \* k\*log(k))，之后进行归并时，复杂度为O(k)，所以总复杂度为O(n/k \* (k\*log(k)+k))= O(n\*logk)。

**问题2：**

对于一个大小不定的数据流，如何设计一个算法，保留k个数据，并使得每个元素被保留的概率相同？

**答：**

假设一共有n个数据，n>>k，对前k个数据进行保留，对k+1个数据，按照k/k+1 的概率保留，如果第k+1个数据保留下来，则从前k个数据中随机抽取一个丢弃，则前k个数据保留下的概率为：1-( (k/k+1) \* (1/k) )= k/k+1 。

之后归纳假设，前i个数据被留下的概率都是 i/m, 则到第i+1个数据时，按照k/i+1保留，同样计算得到，所有元素被保留的概率为k/i+1，可证。

**该问题为：蓄水池抽样问题**

**问题3：**

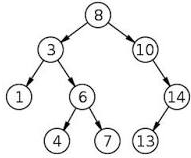
对于一个大小不定的数据流，如何设计一个算法，寻找其下中位数（即如果是n个数据，且n为偶数，则返回第n/2个数据）？

**答：**

错误方法：

维护一颗二叉排序树（根节点比左子树所有节点大，比右子树所有节点小），最后返回根节点。

错误原因：二叉排序树不能保证左右子树节点数相同或者仅差1，所以根节点不是中位数。



正确方法：

维护一个最大堆，一个最小堆，如下图所示：

元素大小

最大堆

最小堆

如图，对于第一个元素，将其放入最大堆中。对于第i个元素，比较其与两个堆堆顶元素的大小关系，如果比最大堆堆顶小，则放入最大堆中，比最小堆堆顶大，则放入最小堆中。介于二者之间，则放入两者元素较少的一个，相等则放在最大堆中。放入之后，如果两个堆元素个数相差2，则将元素个数多的堆的堆顶元素放在少的堆顶，之后进行调整。

取出时，如果两堆元素个数不同，则取个数多的堆的堆顶元素，如果相同，则取最大堆堆顶（下中位数）。