优先队列、堆排序

优先队列

- 优先队列
  - 一种关于集合***S***的数据结构，集合中的元素都有键值***key***
  - 最大优先队列支持以下操作：
    - **INSERT(*S*, *x*)**：把元素***x***插入集合***S***，可表示为***S*←*S*∪{*x*}**
    - **MAXIMUM(*S*)**：返回集合***S***中具有最大键值的元素
    - **EXTRACT-MAX(*S*)**：去掉并返回集合*S*中具有最大键值的元素
    - **INCREASE-KEY(*S*, *x*, *k*)**：将元素***x***的键值增加到***k***，这里要求***k***不能小于***x***的原键值。
  - 类似的，最小优先队列支持以下操作：
    - **INSERT(*S*, *x*)**、**MININUM(*S*)**
    - **EXTRACT-MIN(*S*)**、**DECREASE-KEY(*S*, *x*, *k*)**

二叉堆

- 二叉堆是在一个数组上通过下标间关系维护父子结点关系的一棵几乎满的二叉树

PARENT(*i*)

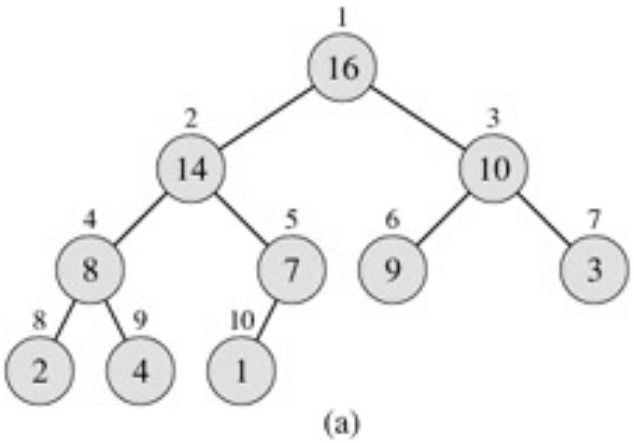
return $\lfloor i/2 \rfloor$

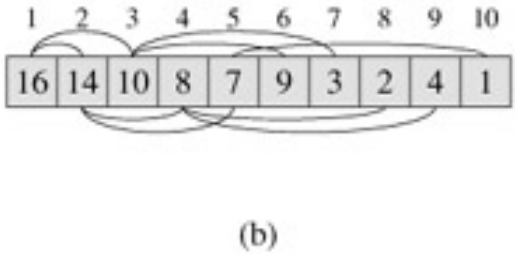
LEFT(*i*)

return $2i$

RIGHT(*i*)

return $2i + 1$





保持堆的性质

MAX-HEAPIFY(*A*, *i*)

1 *l* ← LEFT(*i*)

2 *r* ← RIGHT(*i*)

3 if *l* ≤ *heap-size*[*A*] and *A*[*l*] > *A*[*i*]

4 then *largest* ← *l*

5 else *largest* ← *i*

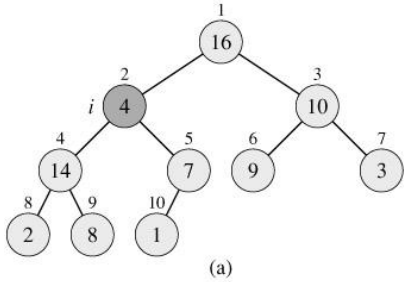
6 if *r* ≤ *heap-size*[*A*] and *A*[*r*] > *A*[*largest*]

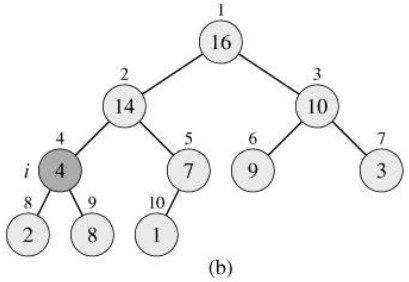
7 then *largest* ← *r*

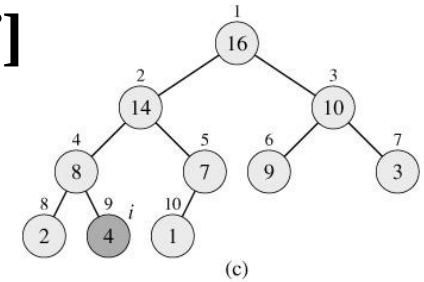
8 if *largest* ≠ *i*

9 then exchange *A*[*i*] ↔ *A*[*largest*]

10 MAX-HEAPIFY(*A*, *largest*)







建堆

BUILD-MAX-HEAP(*A*)

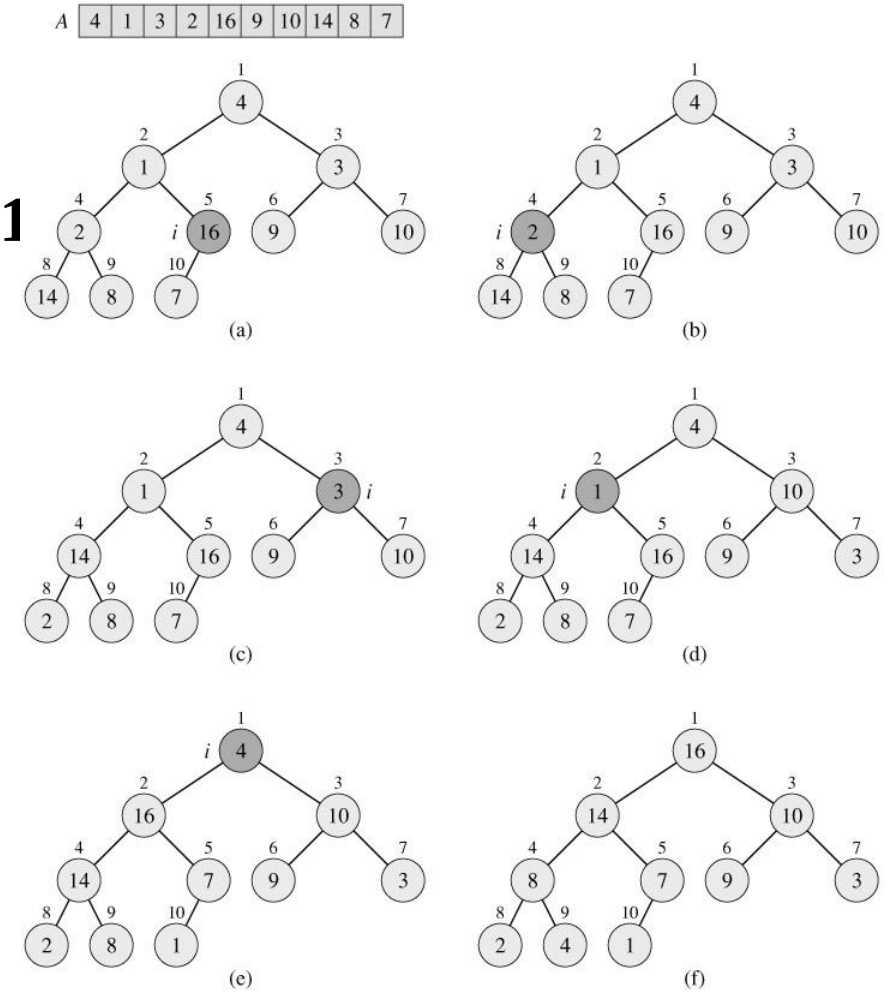
1 *heap-size*[*A*] ← *length*[*A*]

2 for *i* ← [*length*[*A*]/2] downto 1

3 do MAX-HEAPIFY(*A*, *i*)

总运行时间为**O(*n*)**。

合并堆的过程需要用重建堆来实现。



取最大元素 / 取出最大元素

HEAP-MAXIMUM(*A*)

1 return *A*[1]

HEAP-EXTRACT-MAX(*A*)

1 if *heap-size*[*A*] < 1

2 then error "heap underflow"

3 *max* ← *A*[1]

4 *A*[1] ← *A*[*heap-size*[*A*]]

5 *heap-size*[*A*] ← *heap-size*[*A*] - 1

6 MAX-HEAPIFY(*A*, 1)

7 return *max*

增加元素的键值 / 插入新元素

HEAP-INCREASE-KEY(*A*, *i*, *key*)

```
1 if key < A[i]
2     then error "new key is smaller than current key"
3 A[i] ← key
4 while i > 1 and A[PARENT(i)] < A[i]
5     do exchange A[i] ↔ A[PARENT(i)]
6        i ← PARENT(i)
```

MAX-HEAP-INSERT(*A*, *key*)

```
1 heap-size[A] ← heap-size[A] + 1
2 A[heap-size[A]] ← -∞
3 HEAP-INCREASE-KEY(A, heap-size[A], key)
```

堆排序（6.1-1）

- 在高度为h的堆中，最多和最少的元素个数是多少？

练习题6-1：用插入法建堆

- 第6.3节中的BUILD-MAX-HEAP过程也可以通过反复调用MAX-HEAP-INSERT将各元素插入堆中来实现，考虑如下实现：
  - BUILD_MAX_HEAP’(A)
    1 heap-size[A] <- 1
    2 for i <- 2 to length[A]
    3 do MAX-HEAP-INSERT(A, A[i])

(1) 当输入数组相同时，过程BUILD-MAX-HEAP和BUILD-MAX-HEAP’产生的堆是否总是一样的？请给出证明或反例

(2) 证明：在最坏情况下，BUILD-MAX-HEAP’要用Θ(nlgn)时间来建成一个含n个元素的堆。

练习题6-2：d叉堆

- d叉堆与二叉堆很类似，但其中的每个非叶节点有d个子女，而不是两个。
  (1) 如何在一个数组中表示一个d叉堆
  (2) 含n个元素的d叉堆的高度是多少？（用n和d表示）
  (3) 给出d叉最大堆的EXTRACT-MAX的一个有效实现，并用d和n表示出它的运行时间。
  (4) 给出d叉最大堆的INSERT的一个有效实现，并用d和n表示出它的运行时间。
  (5) 给出INCREASE-KEY(A, i, k)的一个有效实现，该过程首先执行A[i] <- max(A[i], k)，并相应地更新d叉最大堆的结构。请用d和n表示出它的运行时间。

思考题

- 在学过Order Statistic的算法之后，对于“包含n个数的集合S中取出前k大的数”的问题可以十分容易地设计出O(n)的算法。但是考虑这样一种场景，有一个庞大的包含整数记录的文件，我们不但无法将文件内容全部读入内存，而且考虑到磁盘读取的效率原因，文件只能允许读一遍。但是k一般是比较小的(k<<n)，k个数（小样本）是可以完全载入内存。在这样一种情境下，设计一个高效的算法，取出大文件中键值最大的k个数。

思考题

- 有整数集合S，初始时包含n个元素。设计一个尽可能优化的算法，支持两种动态操作：

  （1）添加一个元素；

  （2）询问并返回当前集合中元素的中位数。

  给出算法的描述及操作的时间复杂度。