

优先队列、堆排序

优先队列（堆）

- 教材的相关章节
 - Algorithm Design, Chap2.5, 英文版p57
 - Intro to Algo, 2nd edition, Chap6.5, 英文版p138（整个Chap 6都是讲heapsort）
- 维护堆结构的两个基本操作
 - 向上交换、向下交换

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 $S \leftarrow S \cup \{x\}$
 - **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)**、**MINIMUM(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*)

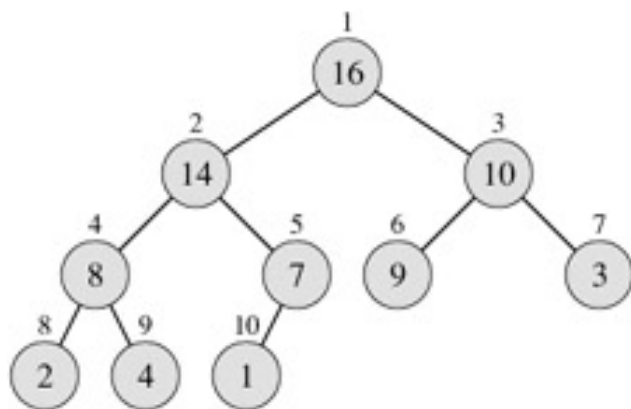
return $\lfloor i/2 \rfloor$

LEFT(*i*)

return $2i$

RIGHT(*i*)

return $2i + 1$



(a)



(b)

保持堆的性质

MAX-HEAPIFY(A, i)

1 $l \leftarrow \text{LEFT}(i)$

2 $r \leftarrow \text{RIGHT}(i)$

3 if $l \leq \text{heap-size}[A]$ and $A[l] > A[i]$

4 then $\text{largest} \leftarrow l$

5 else $\text{largest} \leftarrow i$

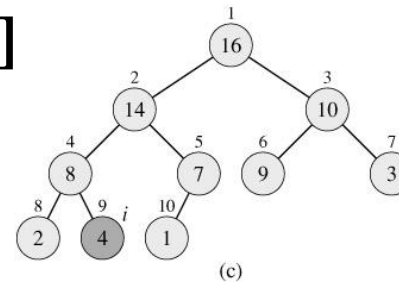
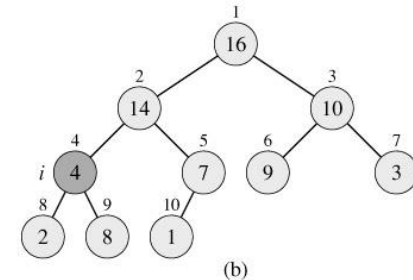
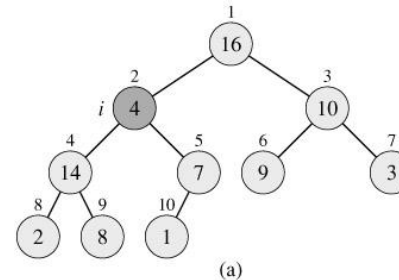
6 if $r \leq \text{heap-size}[A]$ and $A[r] > A[\text{largest}]$

7 then $\text{largest} \leftarrow r$

8 if $\text{largest} \neq i$

9 then exchange $A[i] \leftrightarrow A[\text{largest}]$

10 MAX-HEAPIFY($A, \text{largest}$)



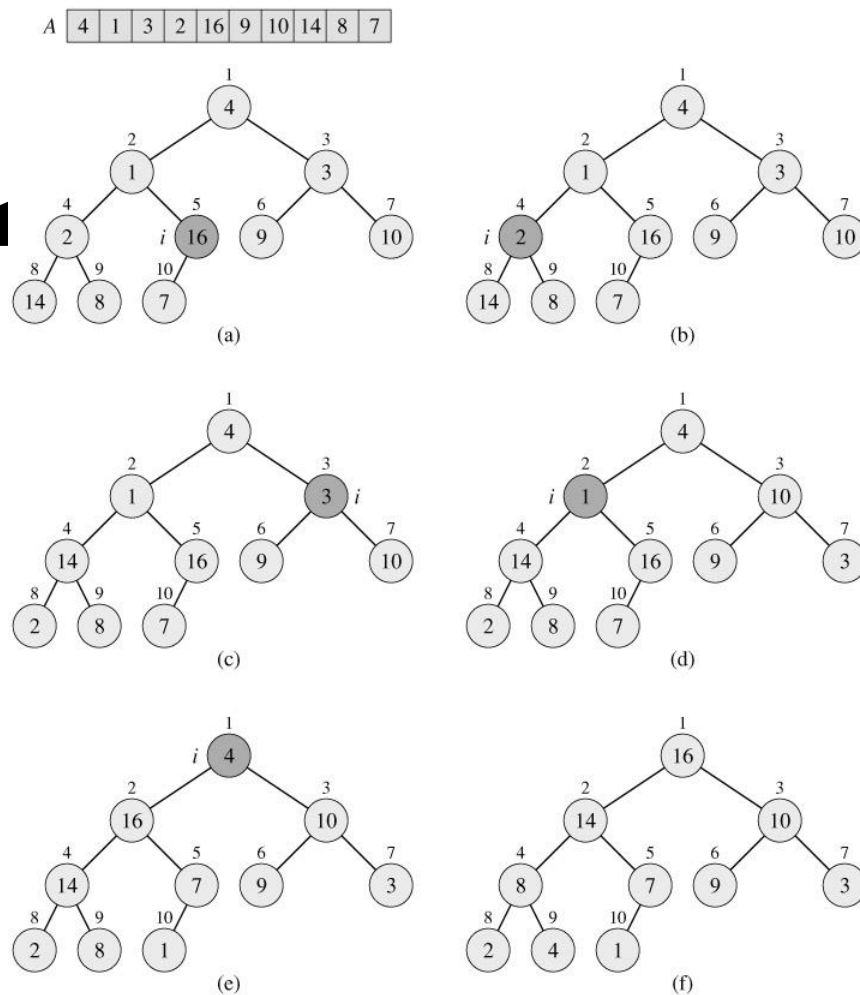
建堆

BUILD-MAX-HEAP(A)

1 $heap-size[A] \leftarrow length[A]$
2 for $i \leftarrow \lfloor length[A]/2 \rfloor$ 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 $max \leftarrow A[1]$

4 $A[1] \leftarrow A[heap-size[A]]$

5 $heap-size[A] \leftarrow heap-size[A] - 1$

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  $A[i] \leftarrow key$ 
4 while  $i > 1$  and  $A[\text{PARENT}(i)] < A[i]$ 
5   do exchange  $A[i] \leftrightarrow A[\text{PARENT}(i)]$ 
6    $i \leftarrow \text{PARENT}(i)$ 
```

MAX-HEAP-INSERT(A, key)

```
1  $heap\text{-}size[A] \leftarrow heap\text{-}size[A] + 1$ 
2  $A[heap\text{-}size[A]] \leftarrow -\infty$ 
3 HEAP-INCREASE-KEY( $A, heap\text{-}size[A], key$ )
```