

# Binary Heap

# Finding parent

PARENT( $i$ )

1 **return**  $\lfloor i/2 \rfloor$

LEFT( $i$ )

1 **return**  $2i$

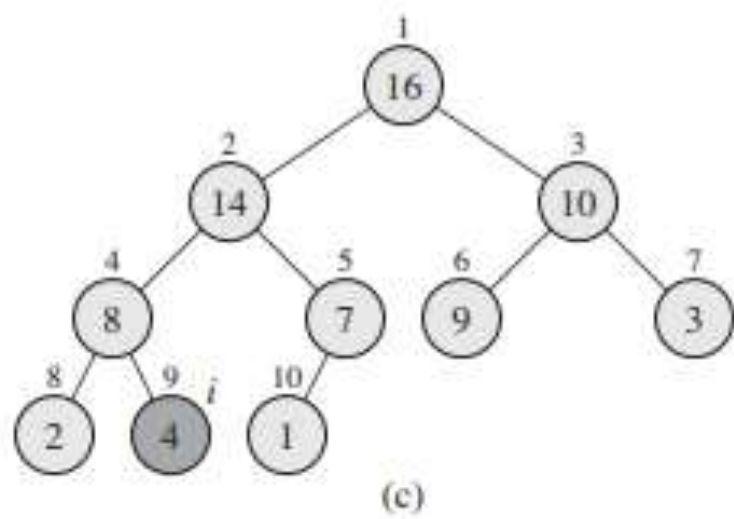
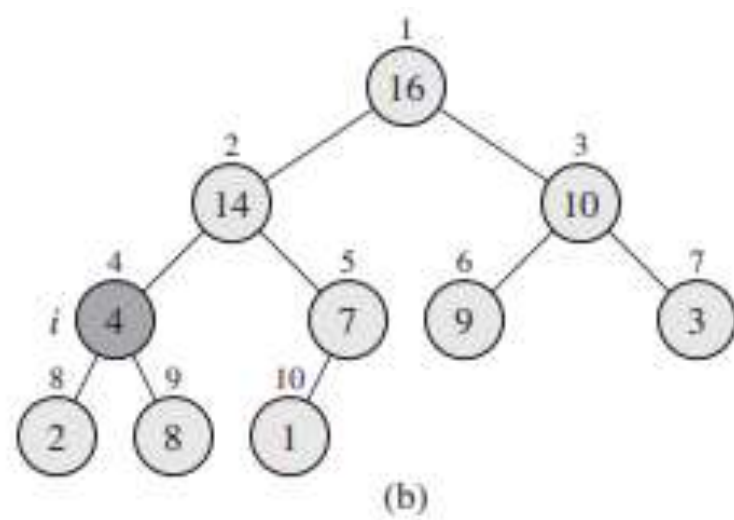
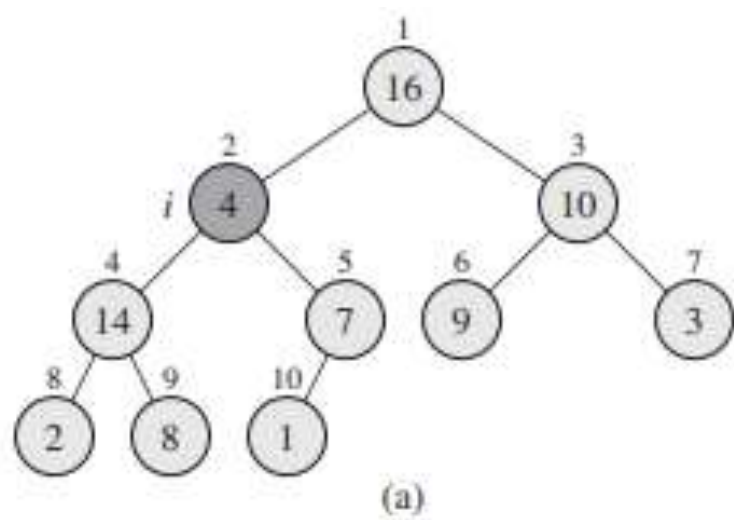
RIGHT( $i$ )

1 **return**  $2i + 1$

# Maintaining the heap property

MAX-HEAPIFY( $A, i$ )

```
1   $l = \text{LEFT}(i)$ 
2   $r = \text{RIGHT}(i)$ 
3  if  $l \leq A.\text{heap-size}$  and  $A[l] > A[i]$ 
4       $\text{largest} = l$ 
5  else  $\text{largest} = i$ 
6  if  $r \leq A.\text{heap-size}$  and  $A[r] > A[\text{largest}]$ 
7       $\text{largest} = r$ 
8  if  $\text{largest} \neq i$ 
9      exchange  $A[i]$  with  $A[\text{largest}]$ 
10     MAX-HEAPIFY( $A, \text{largest}$ )
```

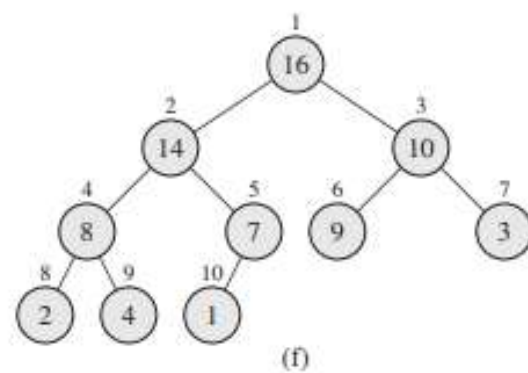
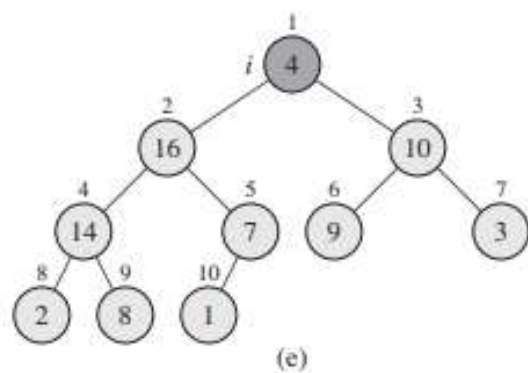
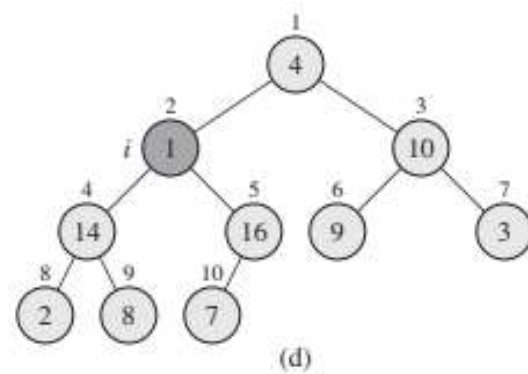
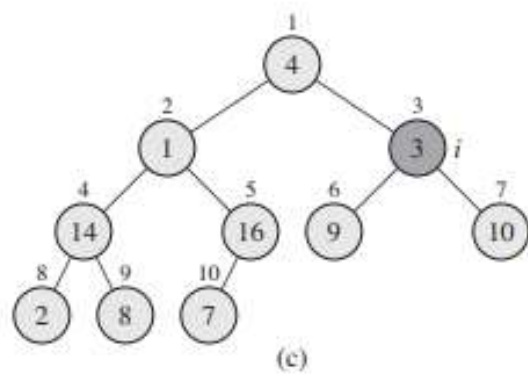
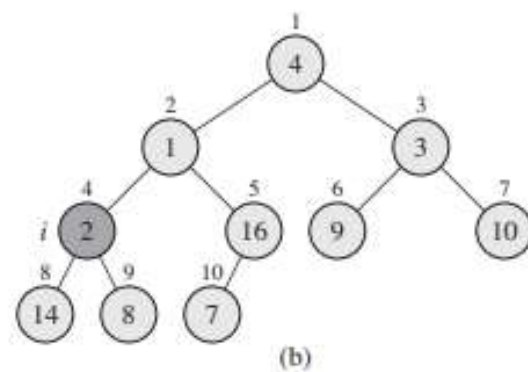
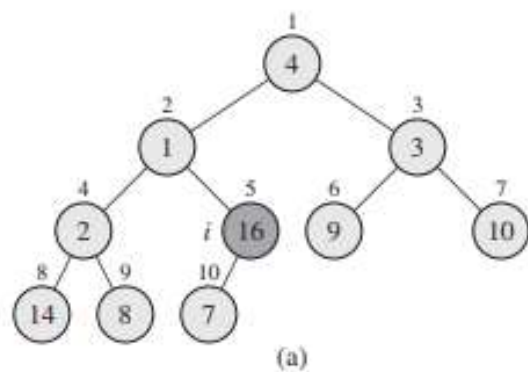


# Building a heap

**BUILD-MAX-HEAP**( $A$ )

```
1   $A.heap-size = A.length$   
2  for  $i = \lfloor A.length/2 \rfloor$  downto 1  
3      MAX-HEAPIFY( $A, i$ )
```

A [ 4 | 1 | 3 | 2 | 16 | 9 | 10 | 14 | 8 | 7 ]



# The heapsort algorithm

HEAPSORT( $A$ )

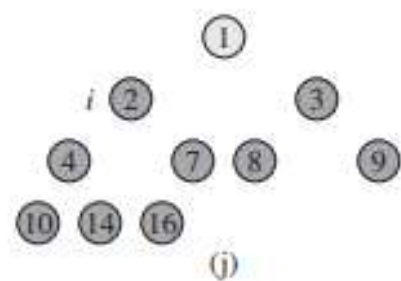
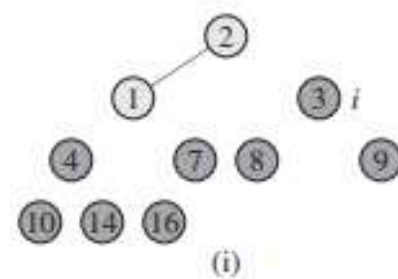
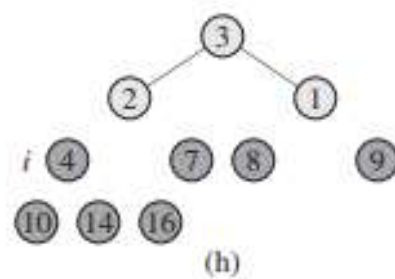
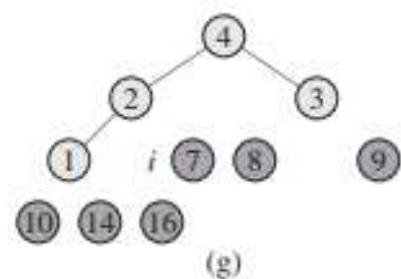
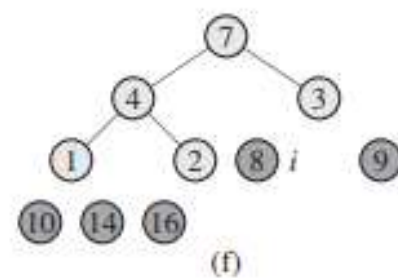
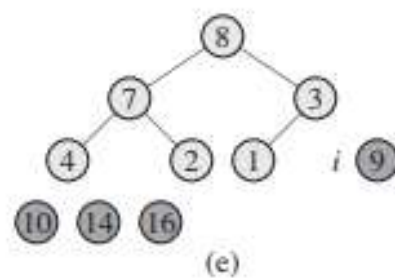
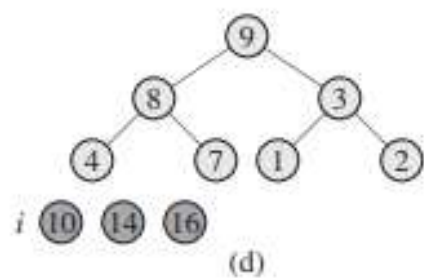
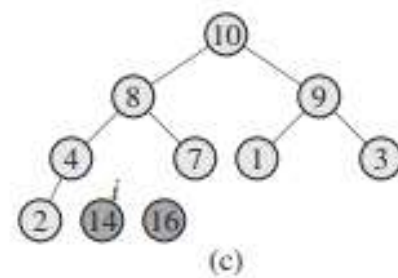
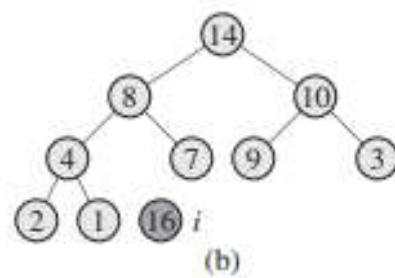
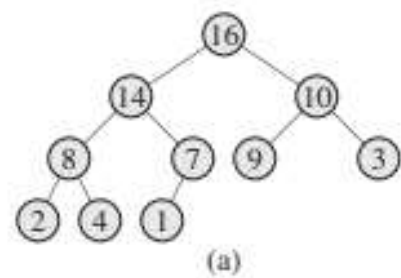
1 BUILD-MAX-HEAP( $A$ )

2 **for**  $i = A.length$  **downto** 2

3     exchange  $A[1]$  with  $A[i]$

4      $A.heap-size = A.heap-size - 1$

5     MAX-HEAPIFY( $A, 1$ )



A 

|   |   |   |   |   |   |   |    |    |    |
|---|---|---|---|---|---|---|----|----|----|
| 1 | 2 | 3 | 4 | 7 | 8 | 9 | 10 | 14 | 16 |
|---|---|---|---|---|---|---|----|----|----|

(k)



# Priority queues

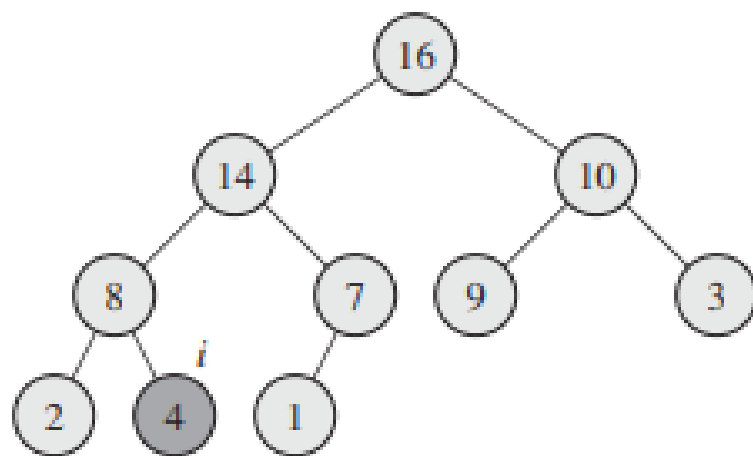
HEAP-EXTRACT-MAX( $A$ )

```
1  if  $A.heap-size < 1$ 
2      error "heap underflow"
3   $max = A[1]$ 
4   $A[1] = A[A.heap-size]$ 
5   $A.heap-size = A.heap-size - 1$ 
6  MAX-HEAPIFY( $A, 1$ )
7  return  $max$ 
```

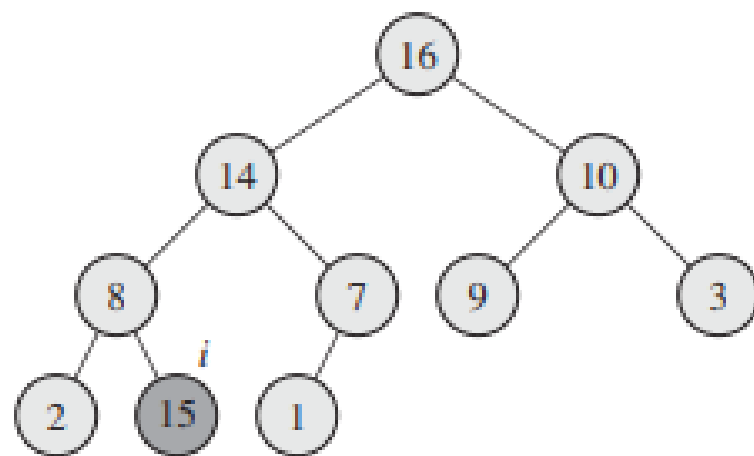
# Increase key

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

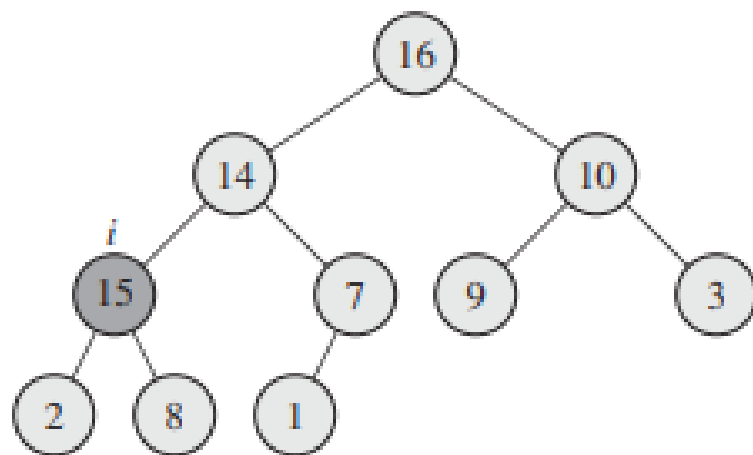
```
1  if  $key < A[i]$ 
2      error "new key is smaller than current key"
3   $A[i] = key$ 
4  while  $i > 1$  and  $A[PARENT(i)] < A[i]$ 
5      exchange  $A[i]$  with  $A[PARENT(i)]$ 
6       $i = PARENT(i)$ 
```



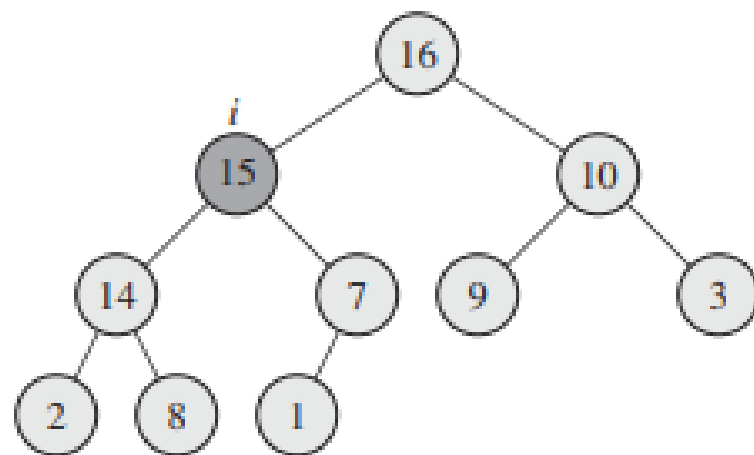
(a)



(b)



(c)



(d)