Heap

DATASTRUCTURES & ALGORITHMS

Definition

The (**binary**) heap is:

- Complete binary tree
- Has heap property

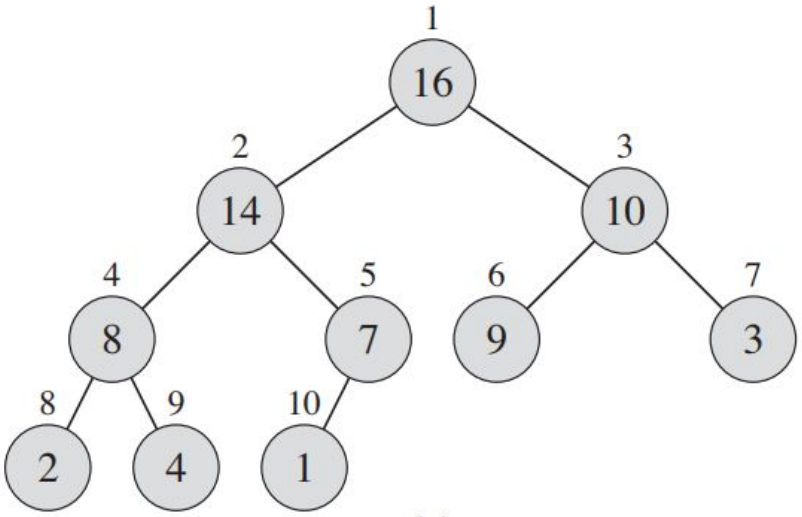
There are two kinds of binary heaps:

- Max-heaps
- Min-heaps

max-heap property is that for every node *i* other than the root:

$$n\_i.parent.key >= n\_i.key$$

Max-heap example

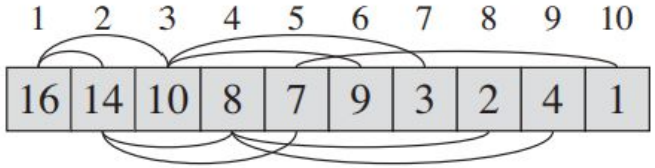


(a)

```
PARENT(i)
1  return ⌊i/2⌋

LEFT(i)
1  return 2i

RIGHT(i)
1  return 2i + 1
```



(b)

Exercise

6.1-1

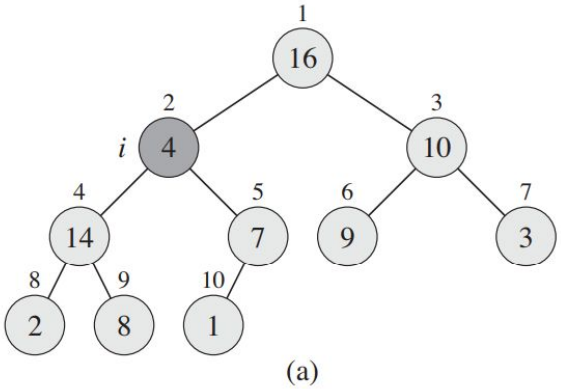
What are the minimum and maximum numbers of elements in a heap of height h?

6.1-2

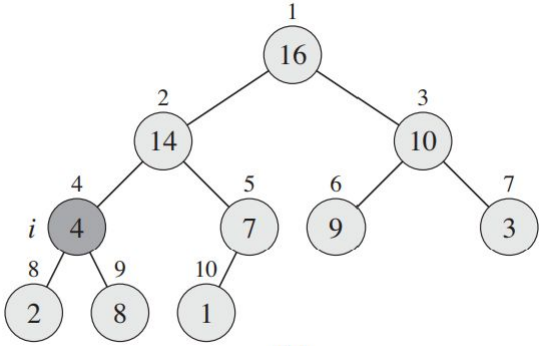
Show that an n-element heap has height $\lfloor \lg n \rfloor$.

Maintaining the heap property

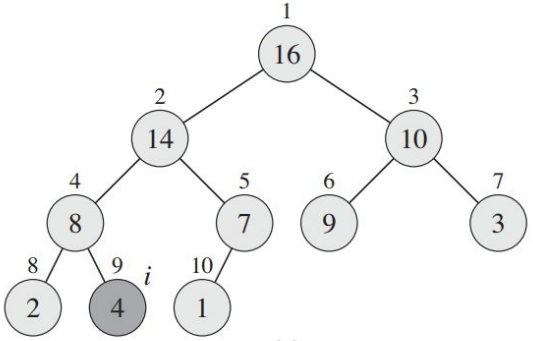
```
MAX-HEAPIFY(A, i)
 1  l = LEFT(i)
 2  r = RIGHT(i)
 3  if l ≤ A.heap-size and A[l] > A[i]
 4       largest = l
 5  else largest = i
 6  if r ≤ A.heap-size and A[r] > A[largest]
 7       largest = r
 8  if largest ≠ i
 9       exchange A[i] with A[largest]
10       MAX-HEAPIFY(A, largest)
```



(a)



(b)



(c)

Running Time of Max_Heapify

What is the Worst case?

$$T(n) \leq T(2n/3) + \Theta(1)$$

$$T(n) = O(\lg n) = O(h)$$

Master Theorem:

$$T(n) = aT(n/b) + f(n)$$

If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$

Building a heap

```
BUILD-MAX-HEAP(A)
1  A.heap-size = A.length
2  for i = ⌊A.length/2⌋ downto 1
3      MAX-HEAPIFY(A, i)
```

Exercise:

6.3-3

Show that there are at most $\lceil n/2^{h+1} \rceil$ nodes of height h in any n-element heap.

Running time of building heap

$$\sum_{h=0}^{\lfloor \lg n \rfloor} \left\lceil \frac{n}{2^{h+1}} \right\rceil O(h) = O\left(n \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h} \right)$$

$$\begin{aligned} \sum_{h=0}^{\infty} \frac{h}{2^h} &= \frac{1/2}{(1-1/2)^2} \\ &= 2 . \end{aligned}$$

$$\begin{aligned} O\left(n \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h} \right) &= O\left(n \sum_{h=0}^{\infty} \frac{h}{2^h} \right) \\ &= O(n) . \end{aligned}$$

Heap sort

```
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)
```

Exercise:

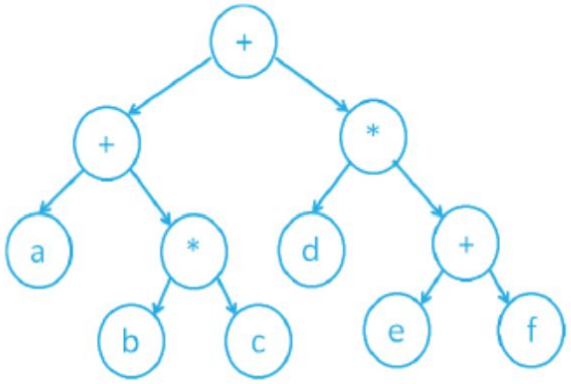
6.4-1

Using Figure 6.4 as a model, illustrate the operation of HEAPSORT on the array $A = \langle 5, 13, 2, 25, 7, 17, 20, 8, 4 \rangle$.

Expression Tree

Expression trees are binary trees with each leaf node serving as an operand and each internal (non-leaf) node serving as an operator.

```
a + (b * c) + d * (e + f)
```



Building expression tree

- Convert expression to postfix
- If a character is an operand, add it to the stack.
- If a character is an operator, pop both values from the stack and make both its children and push the current node again.