Design and Analysis of Algorithms

Lecture-11

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

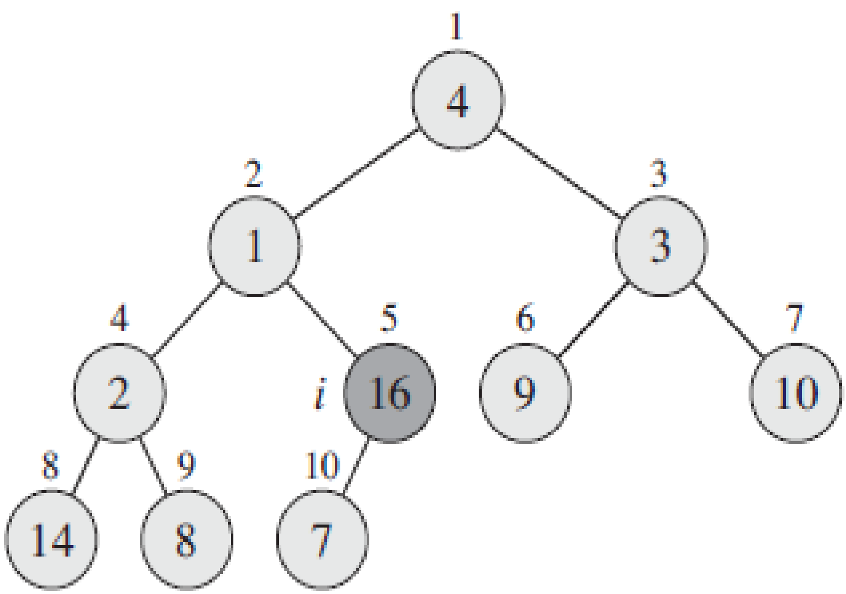
Heapsort

Build-Max-Heap Algorithm

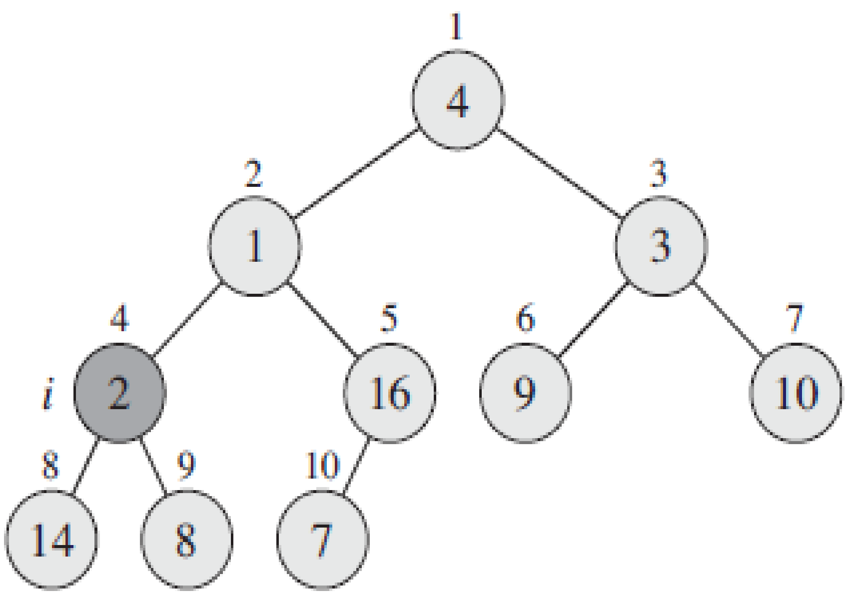
Example: Construct max-heap corresponding to the following elements

4, 1, 3, 2, 16, 9, 10, 14, 8, 7.

Solution:

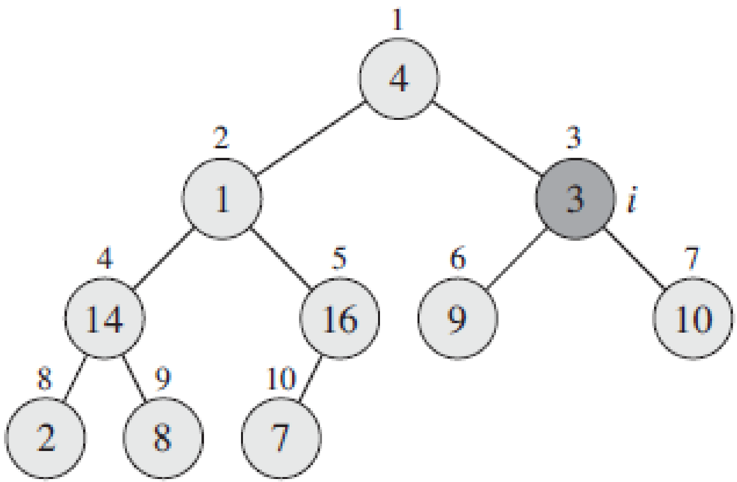


(a)

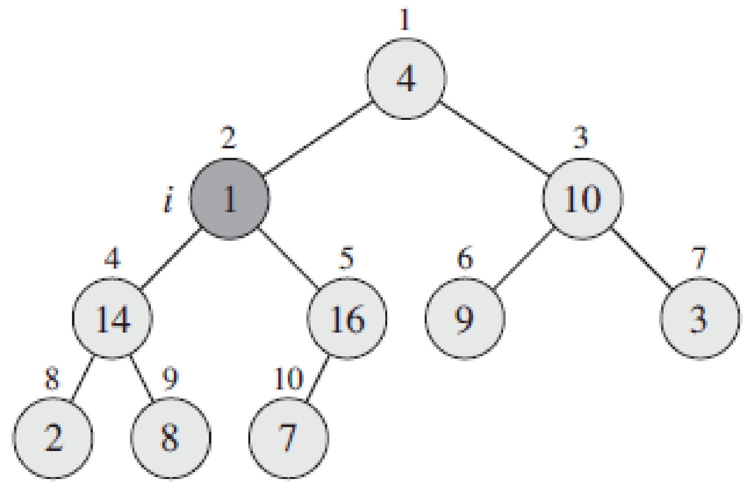


(b)

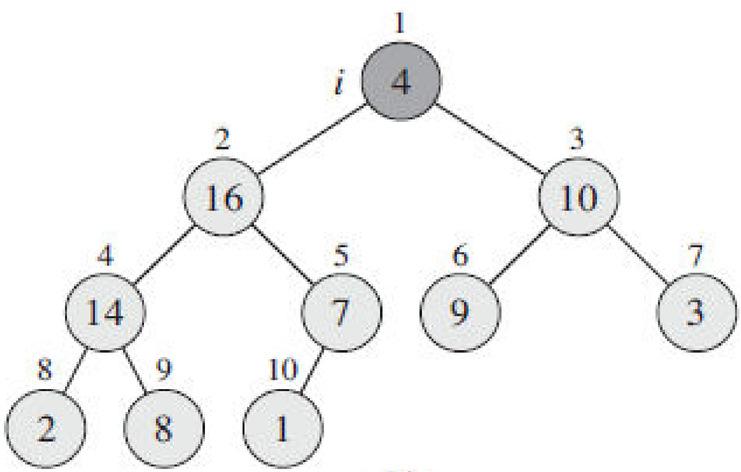
Build-Max-Heap Algorithm (cont.)



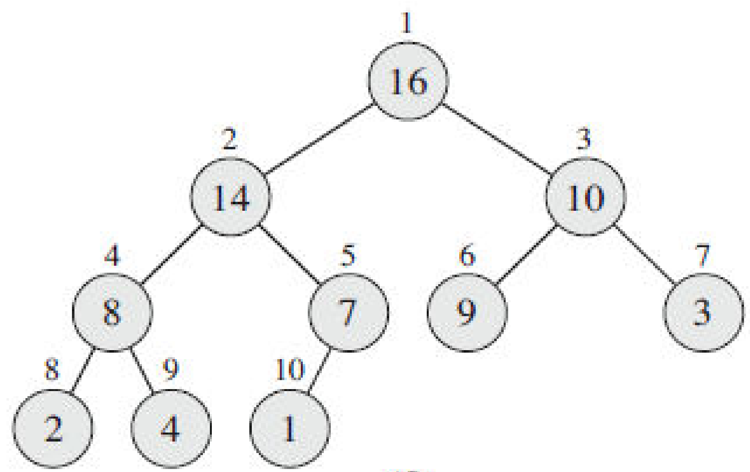
(c)



(d)



(e)



(f)

Heapsort

Build-Max-Heap Algorithm (cont.)

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

The running time of this algorithm is

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

But this upper bound is not asymptotically tight.

Now, we shall calculate tight upper bound.

Heapsort

Time complexity of Build-Max-Heap Algorithm

We can derive a tighter bound by observing that the time for MAX-HEAPIFY to run at a node varies with the height of the node in the tree, and the heights of most nodes are small.

Our tighter analysis relies on the properties that an n-element heap has height $\lfloor \lg n \rfloor$ and at most $\lceil n/2^{h+1} \rceil$ nodes of any height h.

If h is the height of the sub-tree then running time of max-heapify is O(h).

Therefore, total cost of build-max-heap is

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

Heapsort

Time complexity of Build-Max-Heap Algorithm

Now, $T(n) = O\left(n \sum_{h=0}^{\lfloor \lg n \rfloor} \frac{h}{2^h}\right)$

$$= O\left(n \sum_{h=0}^{\infty} \frac{h}{2^h}\right)$$

Since $\sum_{h=0}^{\infty} \frac{h}{2^h} = \frac{1/2}{(1-1/2)^2}$

$$= 2 .$$

Therefore,

$$T(n) = O(2n)$$

$$= \mathbf{O(n)}$$

Heapsort Algorithm

Example: Sort the following elements using heapsort

$$5,\ 13,\ 2,\ 25,\ 7,\ 17,\ 20,\ 8,\ 4.$$

Solution:

Heapsort Algorithm

```
HEAPSORT(A)
    BUILD-MAX-HEAP(A)
    for i = A.length downto 2
        exchange A[1] with A[i]
        A.heap-size = A.heap-size − 1
        MAX-HEAPIFY(A, 1)
```

Time complexity: The HEAPSORT procedure takes time O(nlgn), since the call to BUILD-MAX-HEAP takes time O(n) and each of the n-1 calls to MAX-HEAPIFY takes time O(lgn).

Heapsort Algorithm

Exercise

1. What are the minimum and maximum numbers of elements in a heap of height h?
2. Show that an n-element heap has height $\lfloor \lg n \rfloor$.
3. Is the array with values 23, 17, 14, 6, 13, 10, 1, 5, 7, 12 a max-heap?
4. Show that there are at most $\lceil n/2^{h+1} \rceil$ nodes of height h in any n-element heap.
5. What is the running time of HEAPSORT on an array A of length n that is already sorted in increasing order? What about decreasing order?

Heapsort Algorithm

AKTU Examination Questions

1. Sort the following array using Heap-Sort techniques—
   5,8,3,9,2,10,1,35,22
2. How will you sort following array A of elements using heap sort: A = (23, 9, 18, 45, 5, 9, 1, 17, 6).