HEAP SORT

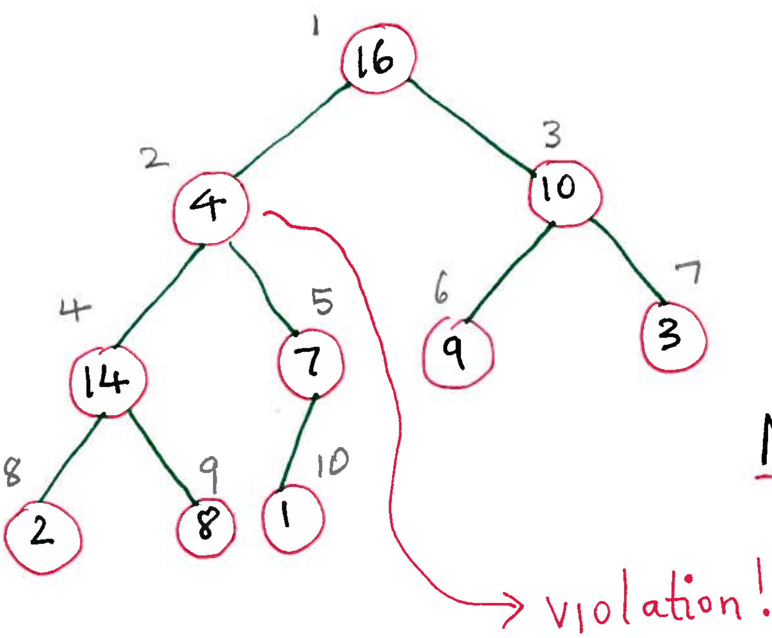
Heaps and MAX-HEAPIFY review

Building a Heap

Heap Sort

Priority Queues (recitation)

Readings : 6.1 – 6.4



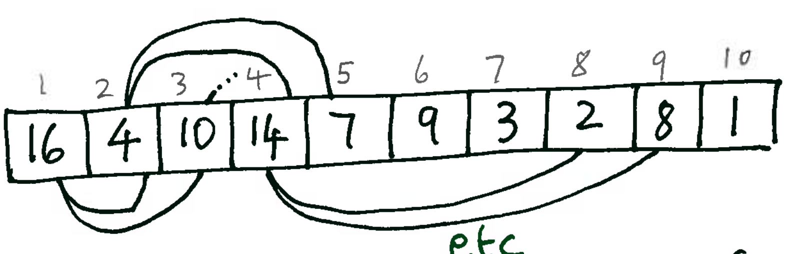
$$\text{Parent}(i) = \left\lfloor \frac{i}{2} \right\rfloor$$

$$\text{Left}(i) = 2i$$

$$\text{Right}(i) = 2i+1$$

Max-heap-property :

$$A[\text{Parent}(i)] \geq A[i]$$



MAX-HEAPIFY (A, 2)

heap-size (A) = 10

$A[2] \Leftrightarrow A[4]$

MAX-HEAPIFY (A, 4)

$A[4] \leftrightarrow A[9]$

O(lg n) time

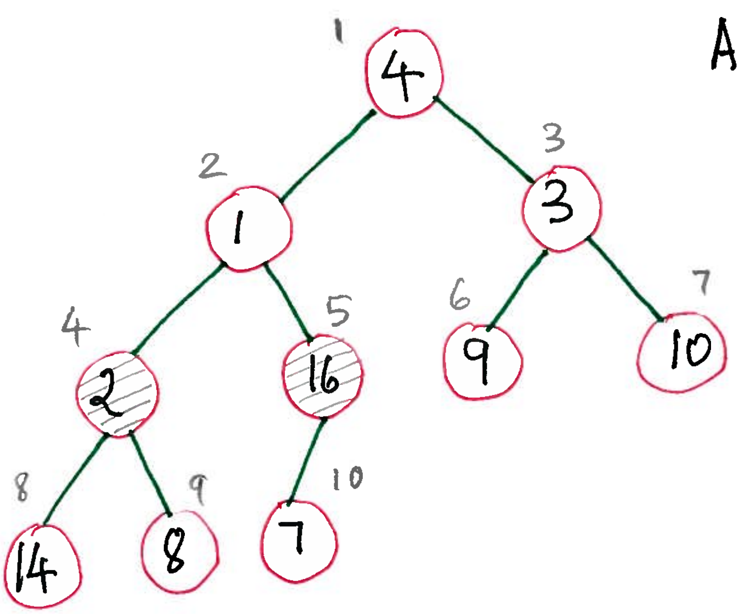
BUILDING A HEAP

$A[1..n]$ converted to a max-heap

Observation: Elements $A[\lfloor \frac{n}{2} + 1 \rfloor \ldots n]$ are all leaves of the tree and can't have children

BUILD-MAX-HEAP (A) :
heap-size(A) = length(A)
O(n) times for $i \leftarrow \lfloor \text{Length}[A]/2 \rfloor$ downto 1
O(lg n) time do MAX-HEAPIFY (A, i)
O(n lg n) overall

EXAMPLE

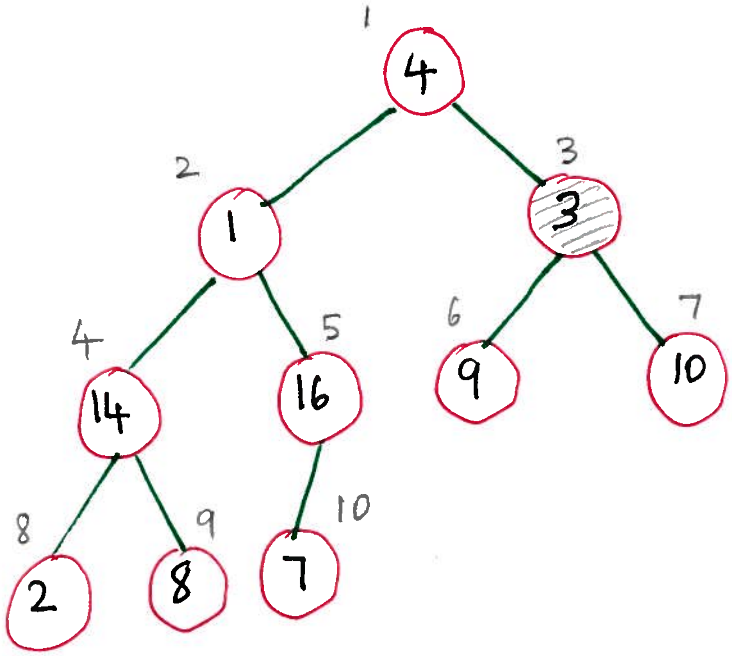


A

4	1	3	2	16	9	10	14	8	7

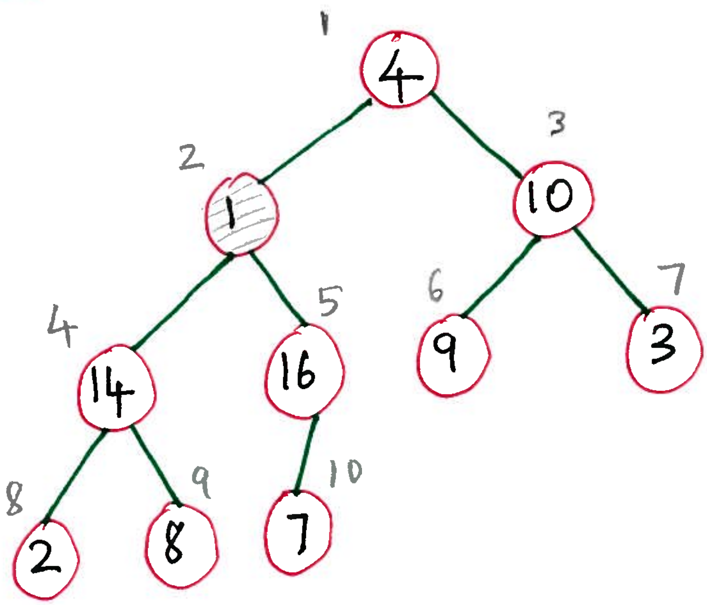
MAX-HEAPIFY (A, 5)
no change

MAX-HEAPIFY (A, 4)
Swap A[4] and A[8]



MAX-HEAPIFY (A, 3)

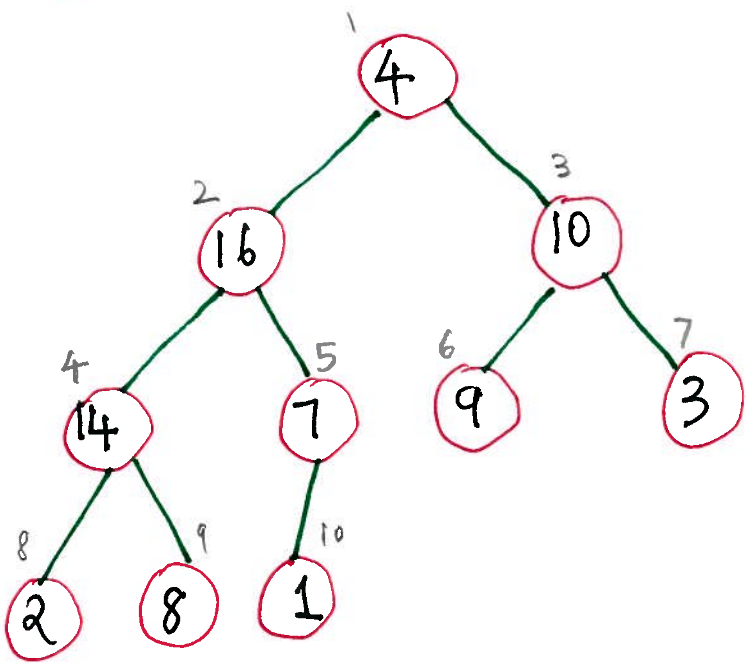
Swap A[3] and A[7]



MAX-HEAPIFY (A, 2)

Swap A[2] and A[5]

Swap A[5] and A[10]



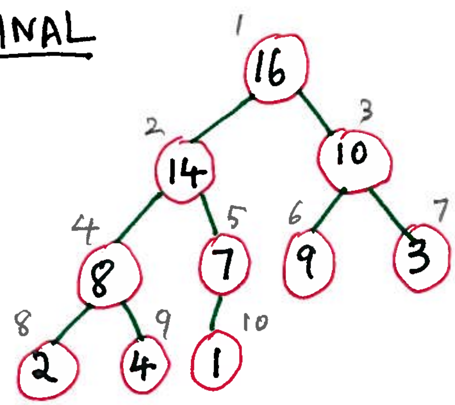
MAX-HEAPIFY (A, 1)

Swap A[1] with A[2]

" A[2] " A[4]

" A[4] " A[9]

FINAL



SORTING STRATEGY

Build max-heap from unordered array

Find maximum element (A[1])

Put it in correct position A[n], A[n] goes to A[1]

Discard node n from heap (decrement heapsize)

new root could violate max-heap property
but children remain max heaps!

HEAP SORT

$O(n\lg n)$ BUILD-MAX-HEAP (A)

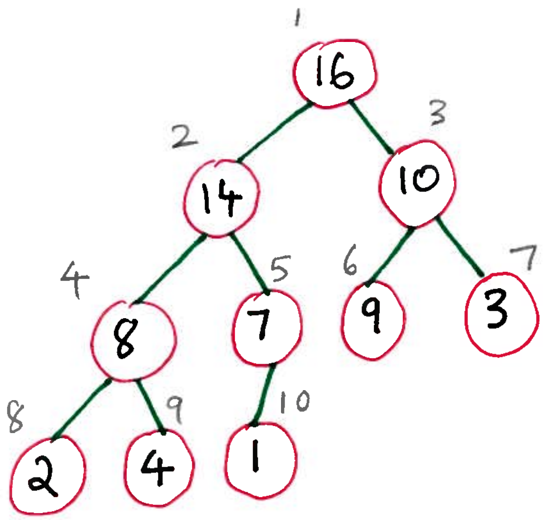
n times for i = length(A) downto 2

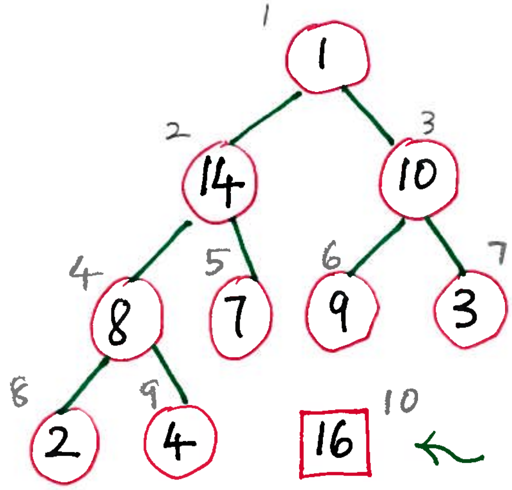
do exchange A[1] ⟷ A[i]

heap-size [A] = heap-size [A] − 1

$O(\lg n)$ MAX-HEAPIFY (A, 1)

$O(n\lg n)$ overall



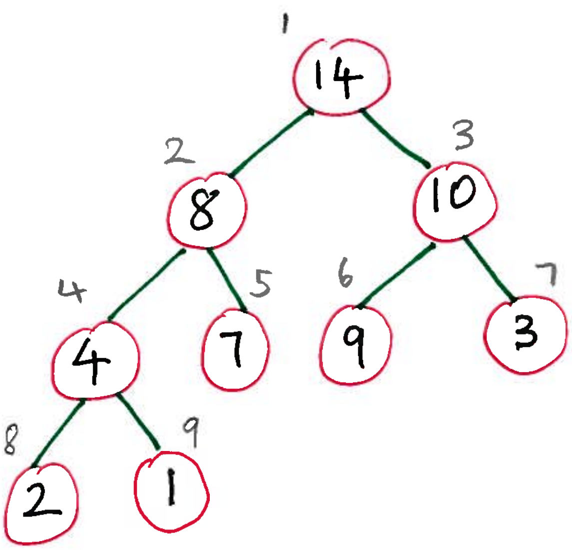


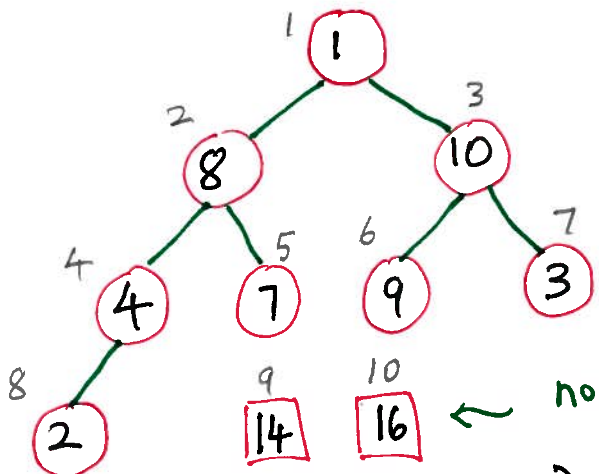
heap-size = 9

MAX-HEAPIFY(A, 1)

not part of heap!

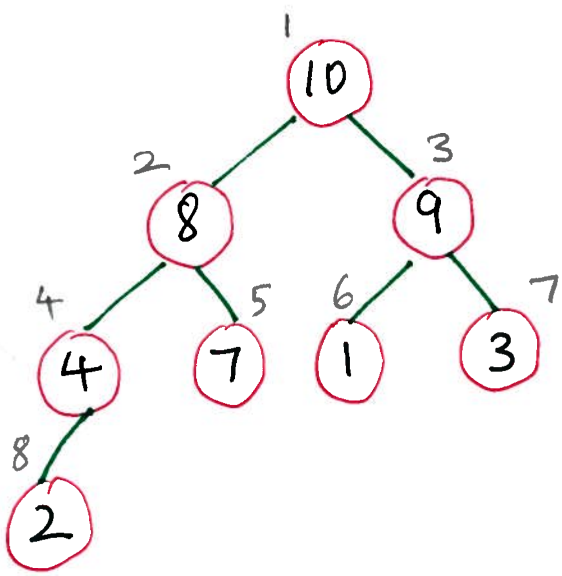
Note: Cannot run MAX-HEAPIFY with heapsize of 10.

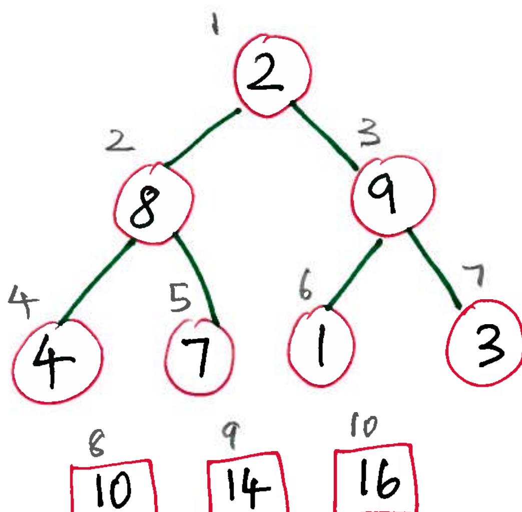




not part of heap

MAX-HEAPIFY (A, 1)





not part of heap

MAX-HEAPIFY (A, 1)

and so on ...

PRIORITY QUEUES.

Abstract Data Type : can be implemented in different ways

INSERT (S, x) : inserts x into set S

MAXIMUM (S) : returns element of S with largest key

EXTRACT-MAX (S) : removes and returns element with largest key

INCREASE-KEY (S, x, k) : increases the value of element x's key to new value k (assumed to be as large as current value)