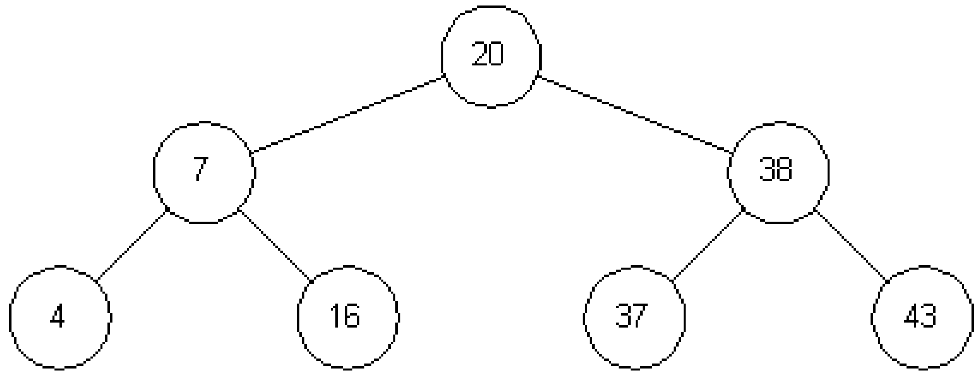
Storing binary trees as arrays



20	7	38	4	16	37	43

Heaps (Max-Heap)

43	16	38	4	7	37	20

43	16	38	4	7	37	20	2	3	6	1	30

HEAP represents a binary tree stored as an array such that:

- Tree is filled on all levels except last
- Last level is filled from left to right
- Left & right child of i are in locations 2i and 2i+1
- HEAP PROPERTY:

Parent value is at least as large as child's value

HeapSort

- First convert array into a heap (BUILD-MAX-HEAP, p133)
- Then convert heap into sorted array (HEAPSORT, p136)

Animation Demos

http://www-cse.uta.edu/~holder/courses/cse2320/lectures/applets/sort1/heapsort.html

http://cg.scs.carleton.ca/~morin/misc/sortalg/

HeapSort: Part 1

$\textsc{Max-Heapify}(\mathit{array}\ A, \mathit{int}\ i)$

$\quad\triangleright$ Assume subtree rooted at i is not a heap;

$\quad\triangleright$ but subtrees rooted at children of i are heaps

$\quad l \leftarrow \textsc{Left}[i]$

$\quad r \leftarrow \textsc{Right}[i]$

$\quad$ **if** $((l \leq \mathit{heap\text{-}size}[A])\ \mathit{and}\ (A[l] > A[i]))$

$\qquad$ **then** $\mathit{largest} \leftarrow l$

$\qquad$ **else** $\mathit{largest} \leftarrow i$

$\quad$ **if** $((r \leq \mathit{heap\text{-}size}[A])\ \mathit{and}\ (A[r] > A[\mathit{largest}]))$

$\qquad$ **then** $\mathit{largest} \leftarrow r$

$\quad$ **if** $(\mathit{largest} \neq i)$

$\qquad$ **then** exchange $A[i] \leftrightarrow A[\mathit{largest}]$

$\qquad\qquad \textsc{Max-Heapify}(A, \mathit{largest})$

O(height of node in location i) = O(log(size of subtree))

p130

HeapSort: Part 2

$\textsc{Build-Max-Heap}(array\ A)$

$heap\text{-}size[A] \leftarrow length[A]$

$\textbf{for } i \leftarrow \lfloor length[A]/2 \rfloor \textbf{ downto } 1$

$\qquad \textbf{do } \textsc{Max-Heapify}(A, i)$

HeapSort: Part 2

BUILD-MAX-HEAP(*array A*)

$heap\text{-}size[A] \leftarrow length[A]$
for $i \leftarrow \lfloor length[A]/2 \rfloor$ **downto** 1
    do MAX-HEAPIFY(A, i)

HEAPSORT(*array A*)

BUILD-MAX-HEAP(A)
for $i \leftarrow length[A]$ **downto** 2
    do exchange $A[1] \leftrightarrow A[i]$
        $heap\text{-}size[A] \leftarrow heap\text{-}size[A] - 1$
        MAX-HEAPIFY$(A, 1)$

O(log n)

Total:
O(nlog n)

Build-Max-Heap Analysis

For the HeapSort analysis, we need to compute:

$$\sum_{h=0}^{\lfloor \log n \rfloor} \frac{h}{2^h}$$

We know from the formula for geometric series that

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

Differentiating both sides, we get

$$\sum_{k=0}^{\infty} kx^{k-1} = \frac{1}{(1-x)^2}$$

Multiplying both sides by x we get

$$\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$$

Now replace $x = 1/2$ to show that

$$\sum_{h=0}^{\lfloor \log n \rfloor} \frac{h}{2^h} \leq \frac{1}{2}$$