Storing binary trees as arrays
**Heaps (Max-Heap)**

**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

Max-Heapify(array A, int i)

▷ Assume subtree rooted at i is not a heap;
▷ but subtrees rooted at children of i are heaps

1. $l \leftarrow \text{LEFT}[i]$
2. $r \leftarrow \text{RIGHT}[i]$
3. if ($l \leq \text{heap-size}[A]$ and ($A[l] > A[i]$))
   then $\text{largest} \leftarrow l$
4. else $\text{largest} \leftarrow i$
5. if ($r \leq \text{heap-size}[A]$ and ($A[r] > A[\text{largest}]$))
   then $\text{largest} \leftarrow r$
6. if ($\text{largest} \neq i$)
   then exchange $A[i] \leftrightarrow A[\text{largest}]
7. $\text{Max-Heapify}(A, \text{largest})$

$O(\text{height of node in location } i) = O(\log(\text{size of subtree}))$
HeapSort: Part 2

\textbf{Build-Max-Heap(array A)}

1. \hspace{1em} heap-size[A] \leftarrow length[A]
2. \hspace{1em} \textbf{for} i \leftarrow \lfloor length[A]/2 \rfloor \hspace{1em} \textbf{downto} \hspace{1em} 1
3. \hspace{1em} \textbf{do} \hspace{1em} \textbf{Max-Heapify}(A, i)
HeapSort: Part 2

Build-Max-Heap(array A)
1  heap-size[A] ← length[A]
2  for i ← ⌈length[A]/2⌉ downto 1
3       do Max-Heapify(A, i)

HeapSort(array A)
1  Build-Max-Heap(A)
2  for i ← length[A] downto 2
4  heap-size[A] ← heap-size[A] − 1
5  Max-Heapify(A, 1)

Total: \(O(n \log n)\)
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

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