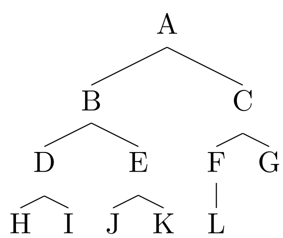
EECS340 - Algorithms - HW#4

Mark Schultz - mxs802

September 30, 2010

6.1-7

The nodes past n/2 are leaves because the children, found at 2n and 2n+1 would be past the end of the array. Therefore they must be leaves because they cannot have any children.



A	B	C	D	E	F	G	H	I	J	K	L
1	2	3	4	5	6	7	8	9	10	11	12
n	n	n	n	n	n	l	l	l	l	l	l

6.3-3

Assume there are $\lceil n/2 \rceil$ leaves with $h = 0$
$f(T, h)$ is the number of nodes with height h in heap T.
$n(T)$ is the number of nodes in heap T
Hypothesis: $f(T, h) \leq \lceil \frac{n(T)}{2^{h+1}} \rceil$
$f(n(T), h-1) \leq \frac{n(T)}{2^h}$ therefore, if the height of a node in T is h then the height of a node in T$'$ is h-1.

$$
\begin{aligned}
f(n(T), h) &= f(n(T'), h-1) \\
&\leq \lceil \frac{n(T')}{2^h} \rceil \\
n(T') &= \lfloor \frac{n(T)}{2} \rfloor \\
&\leq \lceil \frac{\lfloor \frac{n(T)}{2} \rfloor}{2^h} \rceil \\
&\leq \lceil \frac{n(T)}{2^{h+1}} \rceil
\end{aligned}
$$

6.5-6

while $i > 1$ and $A[\textsc{Parent}(i)] < key$ **do**
    $A[i] = A[\textsc{Parent}(i)]$
    $i = \textsc{Parent}(i)$
end while
$A[i] = key$