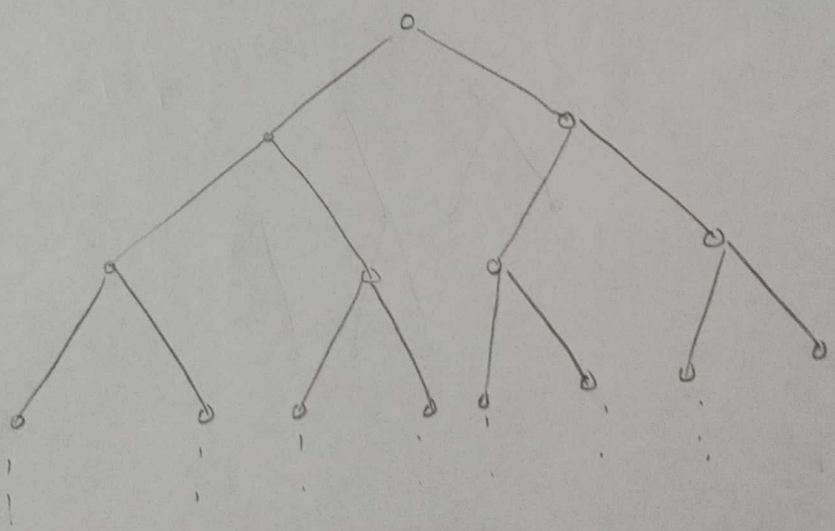
Section 6.1

(1) Since heap = almost-complete binary tree

$\Rightarrow$ complete at all levels except possibly the lowest, which is filled from $L \to R$ upto a point



level	nodes at the level
0	$2^0 = 1$
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
$\vdots$	
h	

so at level h, $\left\{\begin{array}{ll} \text{max possible nodes} : & 2^h \\ \text{min} \quad " \quad : & 1 \end{array}\right\}$

except this level, all levels must be complete.

Total no. of nodes till level $(h-1)$ will be $\left\{\begin{array}{l} \{2^0 + 2^1 + 2^2 + \cdots 2^{h-1}\} \\ = \dfrac{2^h - 1}{2-1} = (2^h - 1) \end{array}\right.$

At level h,

total no of max. no. of nodes $= 2^h - 1 + 2^h = 2^{h+1} - 1$

" min no of nodes $= 2^h - 1 + 1 = 2^h$

$\Rightarrow \quad 2^h \leq \text{Total} \leq 2^{h+1} - 1$

(2) From previous question at height h,

max element $= 2^{h+1}-1$

min " $= 2^h$

$$2^h \le n \le 2^{h+1} - 1$$

$$2^h \le n < 2^{h+1}$$

Both n & h are integers

$$h \le \lg n < h+1$$

$$h = \lfloor \lg n \rfloor$$

From the Definition

$\lfloor x \rfloor = k$

$k \le x < k+1$