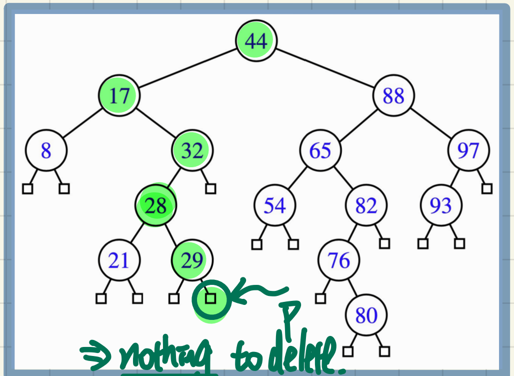
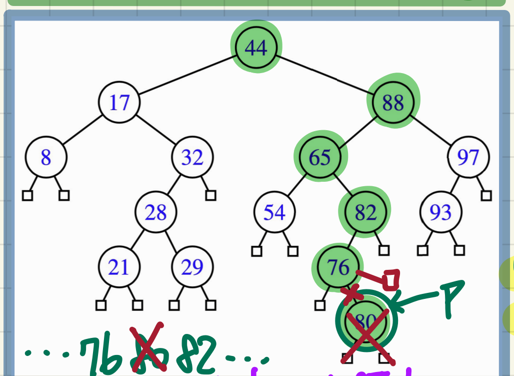
Visualizing BST Operation: Deletion

Case 1: Delete Entry with Key 31



⇒ nothing to delete.

Case 2: Delete Entry with Key 80

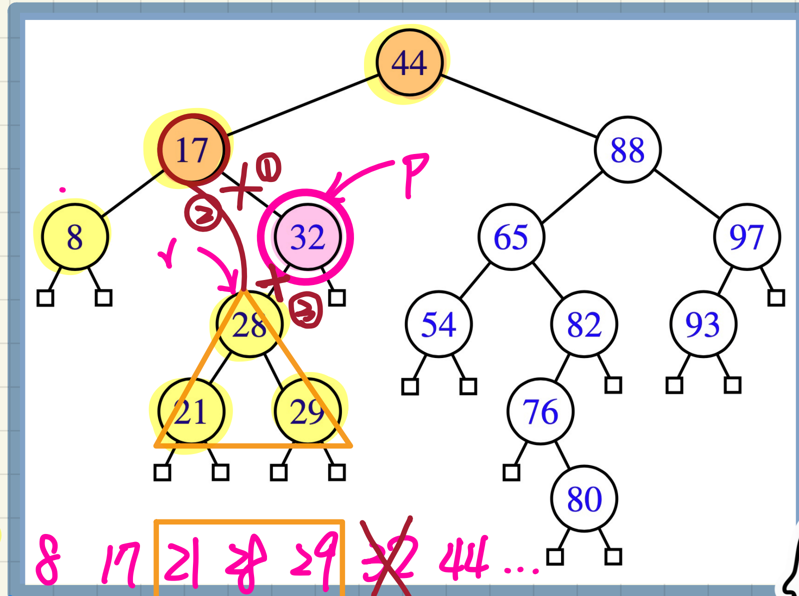


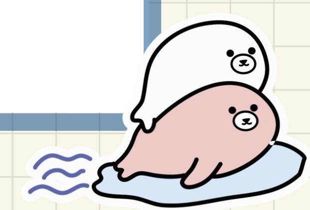
... 76 ~~80~~ 82 ...

⇒ still a BST!

orphan subtree: a subtree rooted at an internal child node

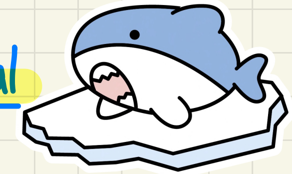
Case 3: Delete Entry with Key 32

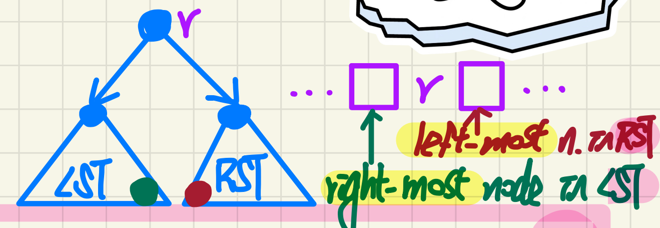




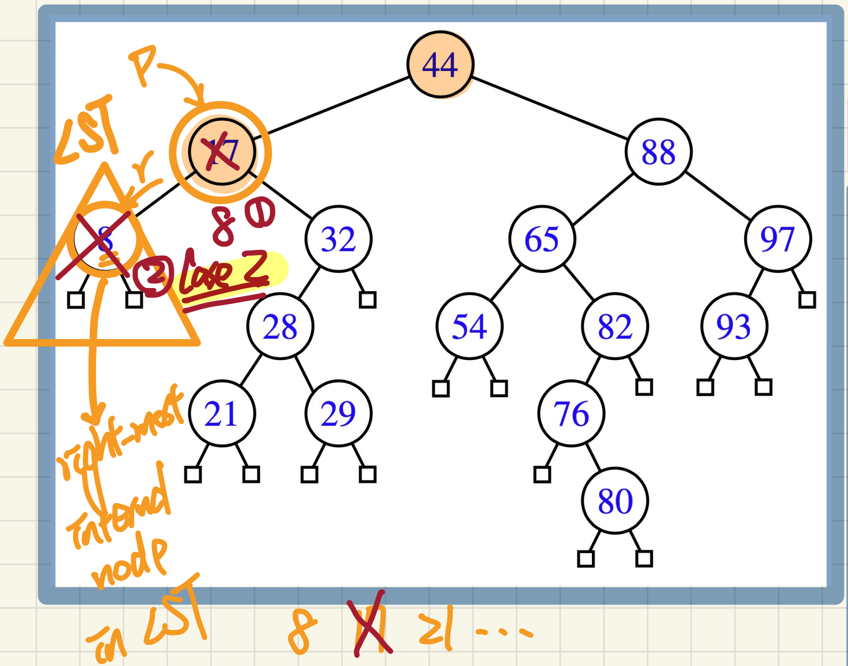
Visualizing BST Operation: Deletion

In-Order Traversal

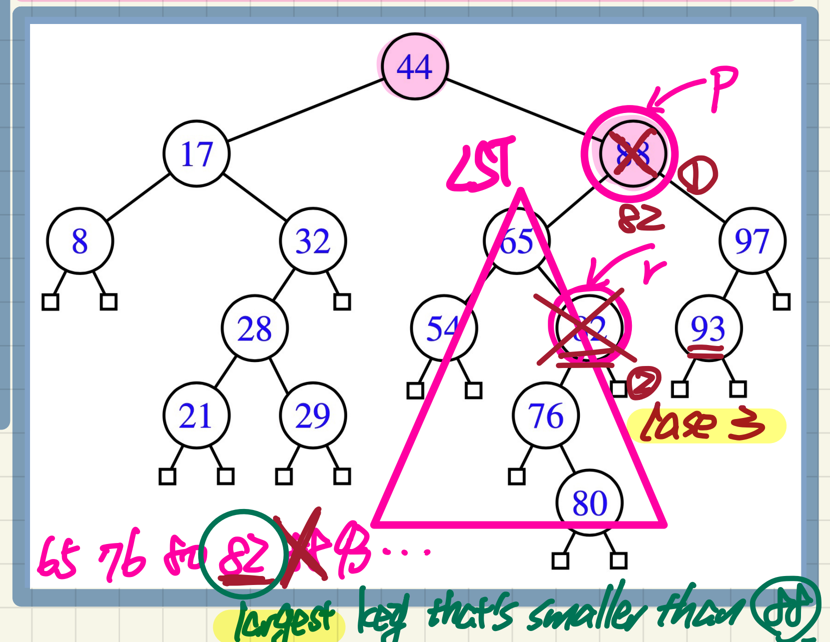




Case 4.1: Delete Entry with Key 17



Case 4.2: Delete Entry with Key 88



Top-Down Heap Construction

Problem: Build a heap out of ***N*** entires, supplied one at a time.

- Initialize an ***empty heap h***.
- As each new entry **e** = (k, v) is supplied, **insert e** into ***h***.

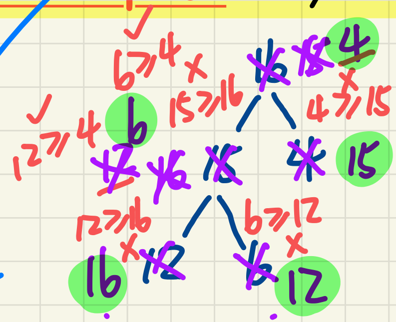
first inserted to level 2

Exercise: Build a heap out of the following 15 keys:

<16, $\boxed{15, 4}$, $\boxed{12, 6, 7, 23}$, 20, 25, 9, 11, 17, 5, 8, 14>

Assumption: Key values supplied one at a time.

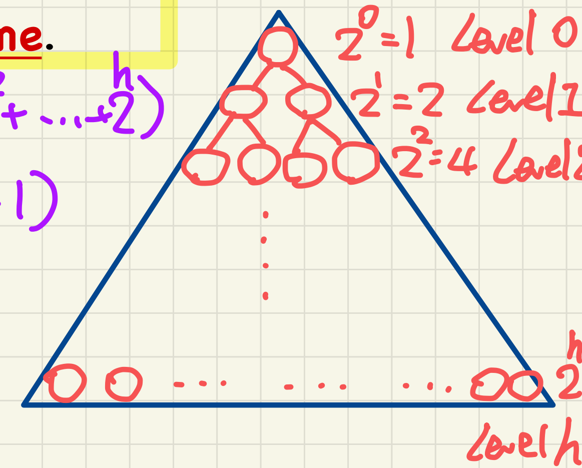
first inserted to level 1



RT: # nodes at level · # upheap bubbling steps

$$* \; \underset{\text{root}}{1} + 2^1 \cdot 1 \leq \log n$$
$$+ 2^2 \cdot 2 \leq \log n$$
$$+ \cdots$$
$$+ 2^h \cdot h \; \log n$$

$$* \leq 1 + \log_2 n \cdot (2^1 + 2^2 + \dots + 2^h)$$
$$= 1 + \log_2 n \, (n-1)$$
$$O(n \cdot \log n)$$



Exercise: Complete inserting the remaining keys to the heap.

Bottom-Up Heap Construction

Problem: Build a heap out of N entires, supplied all at once.

- **Assume**: The resulting heap will be ***completely filled*** at **all** levels.
  $\Rightarrow N = 2^{h+1} - 1$ for some ***height*** $h \geq 1$ $\qquad$ [$h = (log\,(N + 1)) - 1$]
  15. 3.
- Perform the following steps called ***Bottom-Up Heap Construction***:

  Step 1: Treat the first $\boxed{\frac{N+1}{2^1}}$ list entries as heap roots. 8

  $\therefore \frac{N+1}{2^1}$ heaps with height 0 and size $\boxed{2^1 - 1}$ constructed.

  Step 2: Treat the next $\boxed{\frac{N+1}{2^2}}$ list entries as heap roots. 4

  - Each ***root*** sets two heaps from **Step 1** as its ***LST*** and ***RST***.
  - Perform ***down-heap bubbling*** to restore **HOP** if necessary.

  $\therefore \frac{N+1}{2^2}$ heaps, each with height 1 and size $\boxed{2^2 - 1}$, constructed. 3

  Step 3: # entries: ① $\frac{N+1}{2^3}$ 2

  $\frac{16}{2^{3+1}} = 1$

  ...

  Step $h + 1$: Treat next $\boxed{\frac{N+1}{2^{h+1}}} = \frac{(2^{h+1}-1)+1}{2^{h+1}} = 1$ list entry as heap root.

  - Each ***root*** sets two heaps from **Step h** as its ***LST*** and ***RST***.
  - Perform ***down-heap bubbling*** to restore **HOP** if necessary.

  $\therefore \boxed{\frac{N+1}{2^{h+1}} = 1}$ heap, each with height h and size $\boxed{2^{h+1} - 1}$ constructed. $2^{3+1} - 1 = 15$.

② size of each heap: $2^3 - 1 = 7$
③ height of each heap: ②

Exercise: Build a **heap** out of the following 15 keys:

$\langle$ 16, 15, 4, 12, 6, 7, 23, 20, 25, 9, 11, 17, 5, 8, 14 $\rangle$

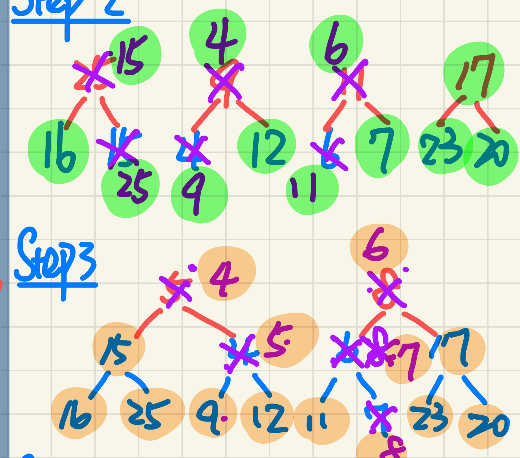
Assumption: Key values supplied all at once.

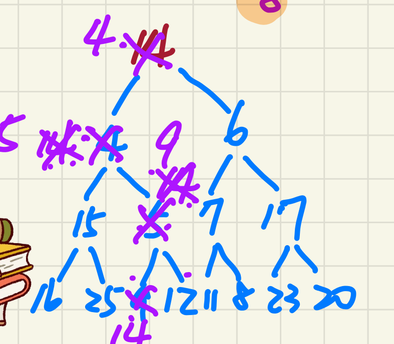
50% Step I
8 heaps, size 1, height 0
16 15 4 12 6 7 23 20

25% Step 2

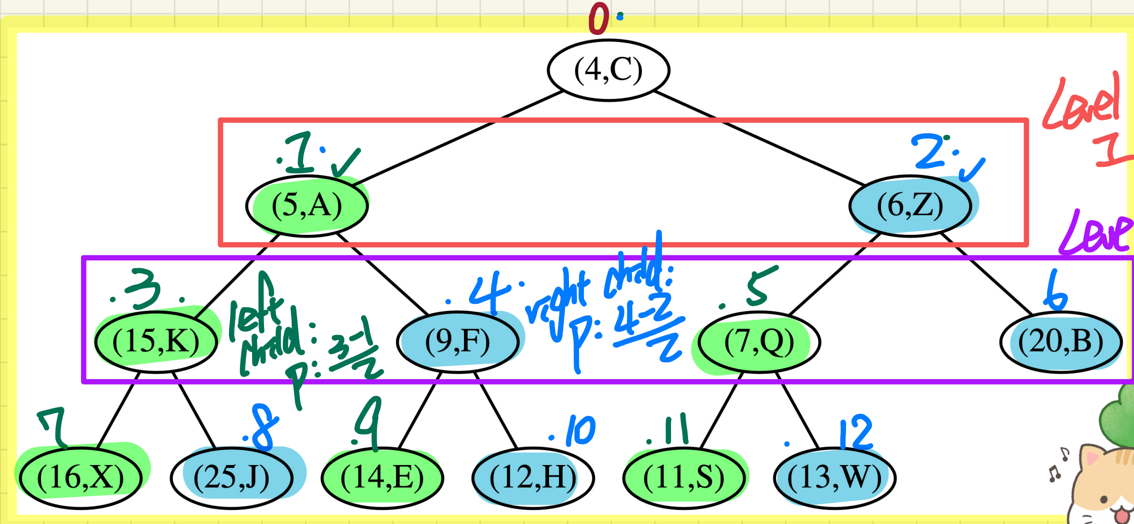
12.5% Step 3

Step





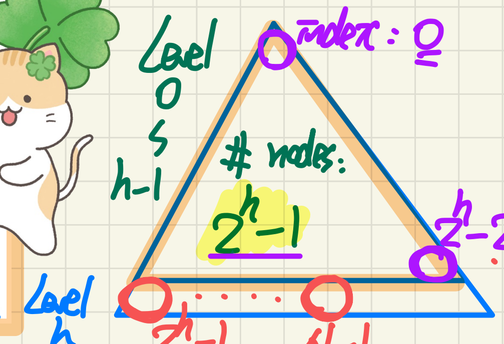
Array-Based Representation of a Complete BT



Exercise

What if the BT is not complete?

(bad for space util.)



$$
index(x) = \begin{cases} 0 & \text{if } x \text{ is the root} \\ 2 \cdot index(\ parent(x)\ ) + 1 & \text{if } x \text{ is a left child} \\ 2 \cdot index(\ parent(x)\ ) + 2 & \text{if } x \text{ is a right child} \end{cases}
$$

L1 L2

0	1	2	3	4	5	6	7	8	9	10	11	12
(4,C)	(5,A)	(6,Z)	(15,K)	(9,F)	(7,Q)	(20,B)	(16,X)	(25,J)	(14,E)	(12,H)	(11,S)	(13,W)