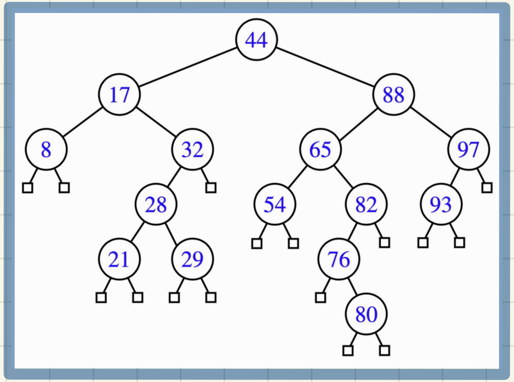
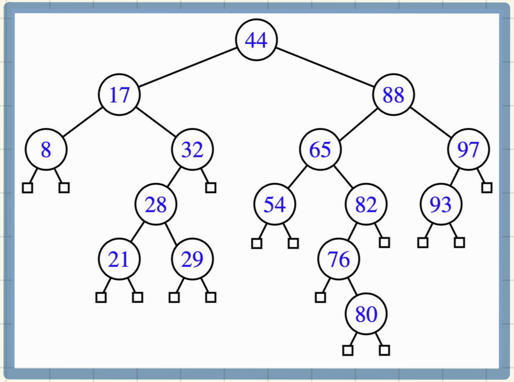
Visualizing BST Operation: Deletion

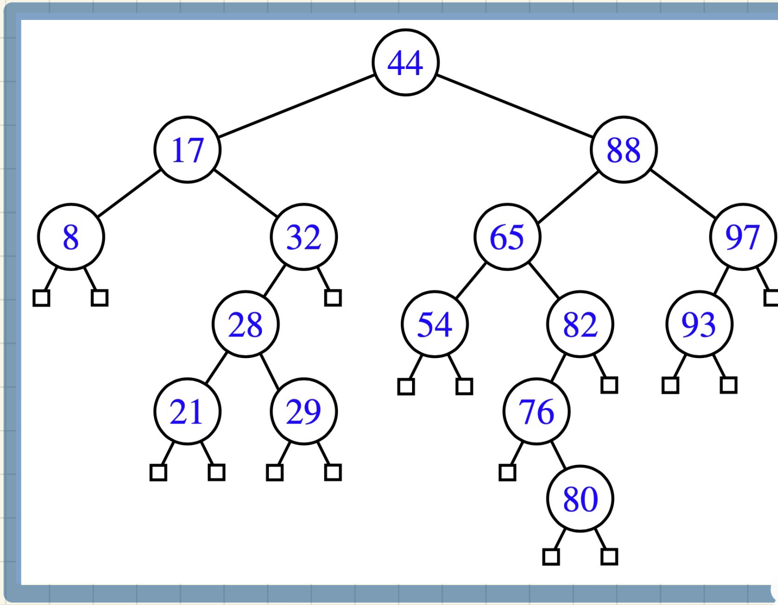
Case 1: Delete Entry with Key 31



Case 2: Delete Entry with Key 80

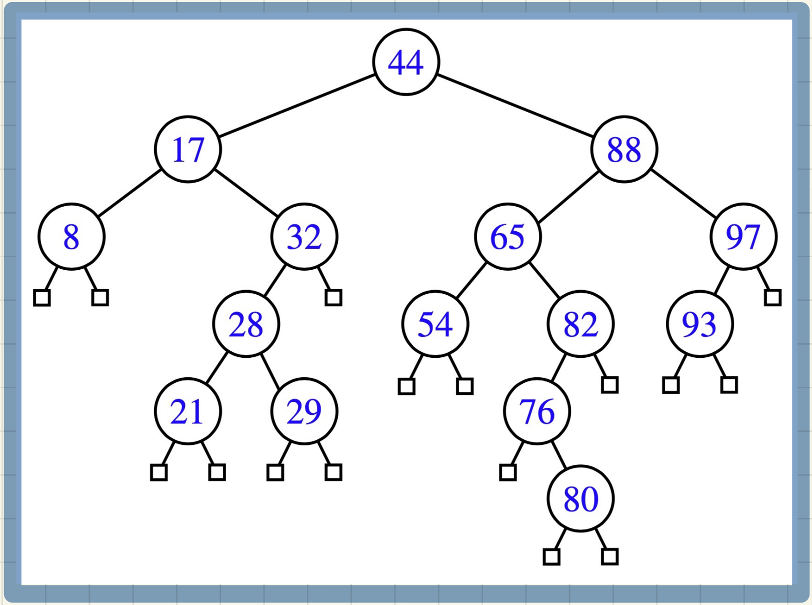


Case 3: Delete Entry with Key 32

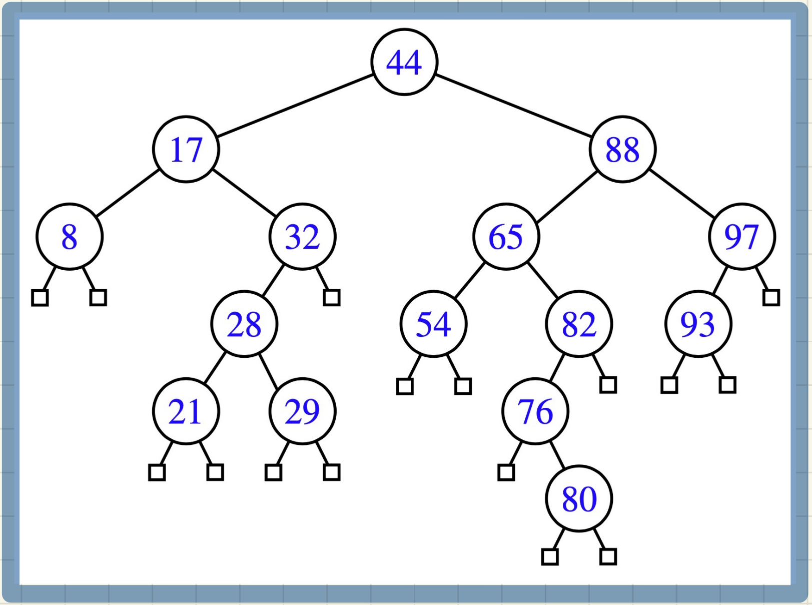


Visualizing BST Operation: Deletion

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***.

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

<16, 15, 4, 12, 6, 7, 23, 20, 25, 9, 11, 17, 5, 8, 14>

Assumption: Key values supplied **one at a time**.

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*** $h \geq 1$ $[\ h = (log\ (N + 1)) - 1\ ]$
- Perform the following steps called *Bottom-Up Heap Construction*:

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

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

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

  ◇ 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 $2^2 - 1$, constructed.

  . . .

  Step $h + 1$: Treat next $\frac{N+1}{2^{h+1}} = \frac{(2^{h+1}-1)+1}{2^{h+1}} = 1$ 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.

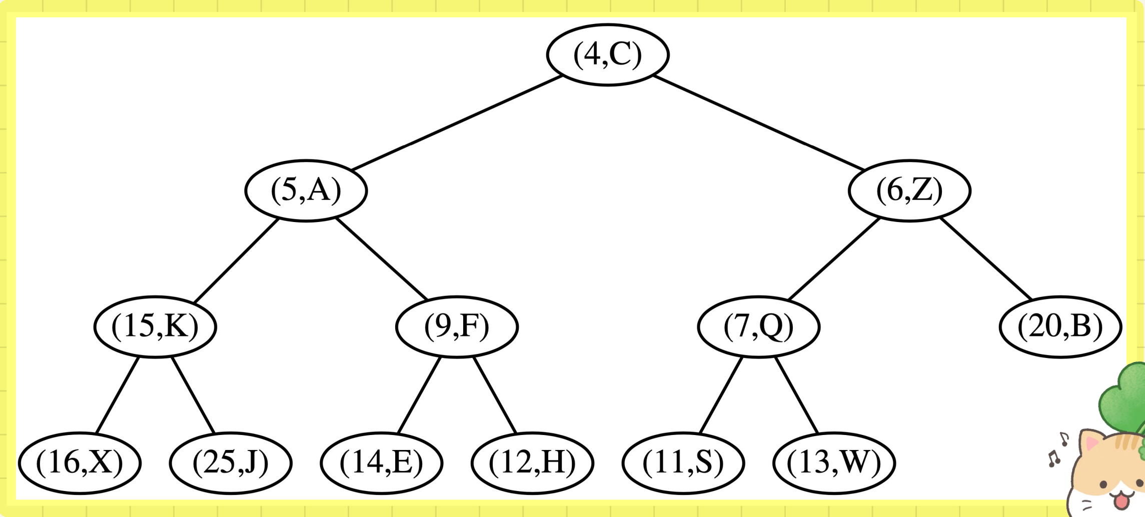
  $\therefore \frac{N+1}{2^{h+1}} = 1$ heap, each with height h and size $2^{h+1} - 1$, constructed.

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

<16, 15, 4, 12, 6, 7, 23, 20, 25, 9, 11, 17, 5, 8, 14>

Assumption: Key values supplied **all at once**.

Array-Based Representation of a Complete BT



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

0	1	2	3	4	5	6	7	8	9	10	11	12