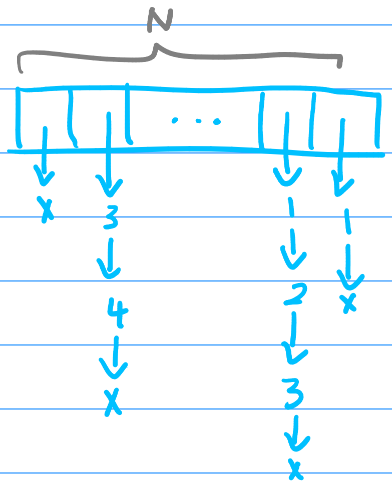
Q2. N slot table, <u>sorted</u> separate chaining for collision resolution.



If we insert $k = 2N$ items, then

- Best case #. of key comps: $N = \frac{k}{2}$
  0 for first N items, N for the rest.

- Avg. #. of key comps per search after this best case: 1.5
  Equally likely for the item to be the 1st or 2nd item per chain, so avg. cost is
  $$\frac{1+2}{2} = 1.5$$

- Worst case #. of key comps: $\sim O(k^2)$
  i^{th} item needs $i-1$ comparisons, so
  $$0+1+2+\ldots+(k-1) = \frac{1}{2}k(k-1) \sim O(k^2)$$

- Avg. #. of key comps after this worst case: $\frac{k}{2}$
  On average, would have to search about half of the k-element chain, so $\frac{k}{2}$ comparisons.

Q3. Hash function: $h(x) = x \bmod 11$

Separate Chaining (sorted):

Values: ~~11~~ ~~16~~ ~~27~~ ~~35~~ ~~22~~ ~~20~~ ~~15~~ ~~24~~ ~~29~~ ~~19~~ ~~13~~

0	1	2	3	4	5	6	7	8	9	10
11 → 22	22	13 → 24 → 35		15	16 → 27		29	19		20

Linear probing:

Values: ~~11~~ ~~16~~ ~~27~~ ~~35~~ ~~22~~ ~~20~~ ~~15~~ ~~24~~ ~~29~~ ~~19~~ ~~13~~

0	1	2	3	4	5	6	7	8	9	10
11	22	35	24	15	16	27	29	19	20	11

Double hashing, where $h_2(x) = (x \bmod 3) + 1$:

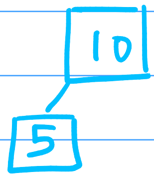
Values: ~~11~~ ~~16~~ ~~27~~ ~~35~~ ~~22~~ ~~20~~ ~~15~~ ~~24~~ ~~29~~ ~~19~~ ~~13~~

0	1	2	3	4	5	6	7	8	9	10
11	13	35	24	22	16	27	15	19	20	29

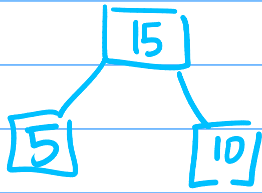
Q8.

insert(10):

insert(5):



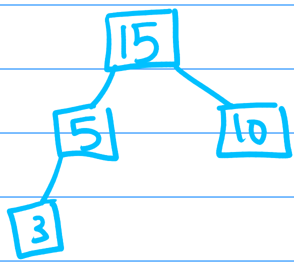
insert(15):



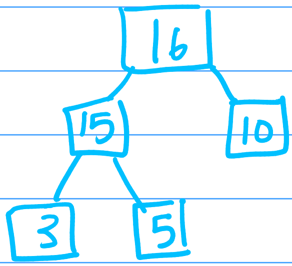
fixup:

$15>10$, swap

insert(3):



insert(16):

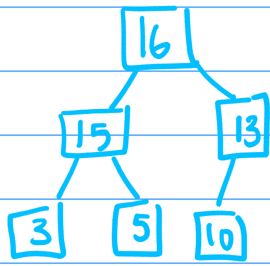


fixup:

$5>16$, swap

$15>16$, swap

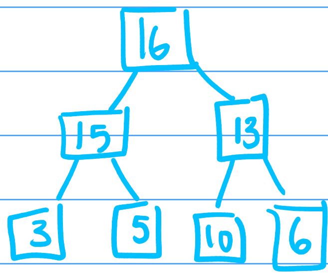
insert(13):



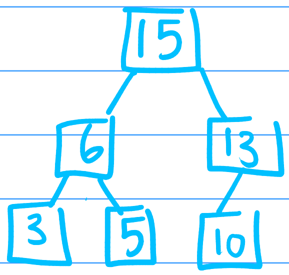
fixup:

$10>13$, swap

insert(6):



delete:

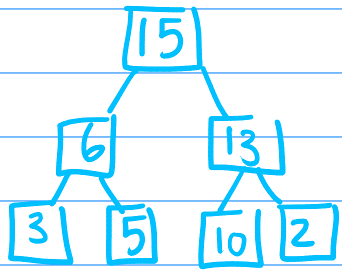


fixdown:

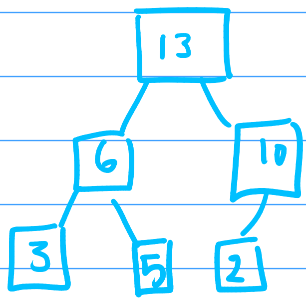
move 6 to root, delete 16

15>6, swap

insert(2):



delete:



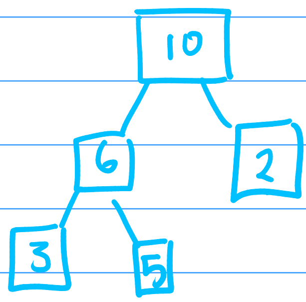
fixdown:

move 2 to root, delete 15

13>2, swap

10>2, swap

delete:



fixdown:

move 2 to the root, delete 15

10>2, swap

Q11. Words: ~~so~~ ~~boo~~ ~~jaws~~ ~~boon~~ ~~boot~~ ~~axe~~ ~~jaw~~ ~~boots~~ sore