DAA HW 3

Pranav Tushar Pradhan N18401944 pp3051@nyu.edu

Udit Milind Gavasane N16545381 umg215@nyu.edu

Rohan Mahesh Gore N19332535 rmg9725@nyu.edu

Q.1.

a) The hash function is $h(k) = k \bmod 9$

Given keys:

5, 28, 19, 15, 20, 33, 12, 17, 10

$h(5) = 5 \bmod 9 = 5$

$h(28) = 28 \bmod 9 = 1$

$h(19) = 28 \bmod 9 = 1$

$h(15) = 15 \bmod 9 = 6$

$h(20) = 20 \bmod 9 = 2$

$h(33) = 33 \bmod 9 = 6$

$h(12) = 12 \bmod 9 = 3$

$h(17) = 17 \bmod 9 = 8$

$h(10) = 10 \bmod 9 = 1$

Final Result:

Slot	Keys (Chaining)
0	
1	$28 \rightarrow 19 \rightarrow 10$
2	20
3	12
4	
5	5
6	$15 \rightarrow 33$
7	
8	17

Q.1.

b) The hash function is $h(K) = K \bmod m$

where $m = 11$.

Given keys:

10, 22, 31, 4, 15, 28, 17, 88, 59

1) $h(K) = h(10) = 10 \bmod 11 = 10 \rightarrow \boxed{\text{Slot } 10}$

2) $h(22) = 22 \bmod 11 = 0 \longrightarrow \boxed{\text{Slot } 0}$

3) $h(31) = 31 \bmod 11 = 9 \longrightarrow \boxed{\text{Slot } 9}$

4) $h(4) = 4 \bmod 11 = 4 \rightarrow \boxed{\text{Slot } 4}$

5) $h(15) = 15 \bmod 11 = 4 \longrightarrow \boxed{\text{slot } 4} \rightarrow$ Collision

$\rightarrow$ place it in next slot since its empty

$15 \rightarrow \boxed{\text{Slot } 5}$

6) $h(28) \rightarrow 28 \bmod 11 = 6 \longrightarrow \boxed{\text{Slot } 6}$

7) $h(17) \rightarrow 17 \bmod 11 = 6 \rightarrow$ collision $\rightarrow$ place in $\boxed{\text{slot } 7}$

8) $h(88) \rightarrow 88 \bmod 11 = 0 \rightarrow$ collision $\rightarrow$ place in $\boxed{\text{Slot } 1}$

9) $h(59) = 59 \bmod 11 = 4 \rightarrow$ collision $\rightarrow$ 4, 5, ~~and~~ 6, 7 occupied, place 59 in $\boxed{\text{slot } 8}$

Final Ans

Slot	0	1	2	3	4	5	6	7	8	9	10
Key	22	88			4	15	28	17	59	31	10

Q1. b.

Quadratic Hashing.

$$h(k, i) = [\,h'(k) + c_1 \cdot i + c_2 \cdot i^2\,] \bmod m$$

where,

$h'(k) = k$, $c_1 = 1$, $c_2 = 3$, $m = 11$

1. Insert 10

$h(10, 0) = [\,10 + 1(0) + 3(0)\,] \bmod 11 = \underline{10 \text{ index}}$

2. Insert 22

$h(22, 0) = [\,22 + 1(0) + 3(0)\,] \bmod 11 = \underline{0 \text{ index}}$

0	22
1	
2	88
3	17
4	4
5	
6	28
7	59
8	15
9	31
10	10

3. Insert 31

$h(31, 0) = [\,31 + 1(0) + 3(0)\,] \bmod 11 = \underline{9 \text{ index}}$

4. Insert 4

$h(4, 0) = [\,4 + 1(0) + 3(0)\,] \bmod 11 = \underline{4 \text{ index}}$

5. Insert 15

$h(15, 0) = [\,15 + 1(0) + 3(0)\,] \bmod 11 = 4 \text{ index}$

But 4th index is occupied

$\therefore\ h(15, 1) = [\,15 + 1(1) + 3(1^2)\,] \bmod 11 = 19 \bmod 11 = \underline{8 \text{ index}}$

6. Insert 28

$h(28,0) = [28 + 1(0) + 3(0)] \bmod 11 = \underline{6 \text{ index}}$

7. Insert 17

$h(17,0) = [17 + 1(0) + 3(0)] \bmod 11 = 6$ index

6th occupied

$\therefore h(17,1) = [17 + 1(1) + 3(1)] \bmod 11 = 10$ index

but 10th occupied

$\therefore h(17,2) = [17 + 1(2) + 3(4)] \bmod 11 = 9$ index

but 9th occupied

$\therefore h(17,3) = [17 + 1(3) + 3(9)] \bmod 11 = \underline{3 \text{ index}}$

8. Insert 88

$h(88,0) = [88 + 1(0) + 3(0)] \bmod 11 = 0$ index X

$h(88,1) = [88 + 1(1) + 3(1)] \bmod 11 = 4$ index X

$h(88,2) = [88 + 1(2) + 3(4)] \bmod 11 = 3$ index X

$h(88,3) = [88 + 1(3) + 3(9)] \bmod 11 = 8$ index X

$h(88,4) = [88 + 1(4) + 3(16)] \bmod 11 = 8$ index X

$h(88,5) = [88 + 1(5) + 3(25)] \bmod 11 = 3$ index X

$h(88,6) = [88 + 1(6) + 3(36)] \bmod 11 = 4$ index X

$h(88,7) = [88 + 1(7) + 3(49)] \bmod 11 = 0$ index X

$h(88,8) = [88 + 1(8) + 3(64)] \bmod 11 = \underline{2 \text{ index}}$

9. Insert 59

$h(59,0) = 4$ index X $\quad h(59,1) = 8$ index X

$h(59,2) = [59 + 1(2) + 3(4)] \bmod 11 = \underline{7 \text{ index}}$

Q1 - b.

Double Hashing.

$$h(k,i) = [h_1(k) + i \cdot h_2(k)] \bmod m$$

where,

$h_1(k) = k$

$h_2(k) = 1 + [k \bmod (m-1)]$

$\therefore$ given $m = 11$

0	22
1	
2	59
3	17
4	4
5	15
6	28
7	88
8	
9	31
10	10

1. Insert 10

$h_1(10) = 10$

$h_2(10) = 1$

$h(k,i) = [10 + 0(1)] \bmod 11$

$k = 10$, $i = 0$

$= 10 \bmod 11$

$= \underline{10 \text{ index}}$

2. Insert 22

$h_1(22) = 22$

$h_2(22) = 1 + [22 \bmod 10] = 3$

$h(k,i) = [22 + 0(3)] \bmod 11$

$= 22 \bmod 11$

$= \underline{0 \text{ index}}$

3. Insert 31

$h_1(31) = 31$

$h_2(31) = 1 + [31 \bmod 10] = 1 + 1 = 2$

$h(31, 0) = [31 + 0(2)] \bmod 11$

$= 31 \bmod 11$

$= 9$ index

4. Insert 4

$h_1(4) = 4$

$h_2(4) = 1 + [4 \bmod 10] = 1 + 4 = 5$

$h(4, 0) = [4 + 0(5)] \bmod 11 = 4$ index.

5. Insert 15

$h_1(15) = 15$

$h_2(15) = 1 + [15 \bmod 10] = 1 + 5 = 6$

$h(15, 0) = [15 + 0(6)] \bmod 11 = 15 \bmod 11 = 4$ index

4th index is occupied

$\therefore h(15, 1) = [15 + 1(6)] \bmod 11 = 21 \bmod 11 = 10$ index

10th index is occupied

$\therefore h(15, 2) = [15 + 2(6)] \bmod 11 = 27 \bmod 11 = 5$ index

6. Insert 28

$h_1(28) = 28$

$h_2(28) = 1 + [28 \bmod 10] = 1 + 8 = 9$

$h(28,0) = [28 + 0(9)] \bmod 11 = 28 \bmod 11 = 6$ index

7. Insert 17

$h_1(17) = 17$

$h_2(17) = 1 + [17 \bmod 10] = 1 + 7 = 8$

$h(17,0) = [17 + 0(8)] \bmod 11 = 17 \bmod 11 = 6$ index

6th index → occupied

$\therefore h(17,1) = [17 + 1(8)] \bmod 11 = 25 \bmod 11 = 3$ index

8. Insert 88

$h_1(88) = 88$

$h_2(88) = 1 + [88 \bmod 10] = 1 + 8 = 9$

$h(88,0) = [88 + 0(9)] \bmod 11 = 88 \bmod 11 = 0$ index

0th index is occupied

$\therefore h(88,1) = [88 + 1(9)] \bmod 11 = 97 \bmod 11 = 9$ index

9th index is occupied

$\therefore h(88,2) = [88 + 2(9)] \bmod 11 = 106 \bmod 11 = 7$ index

9. Insert 59

$h_1(59) = 59$ $h_2(59) = 1 + 9 = 10.$

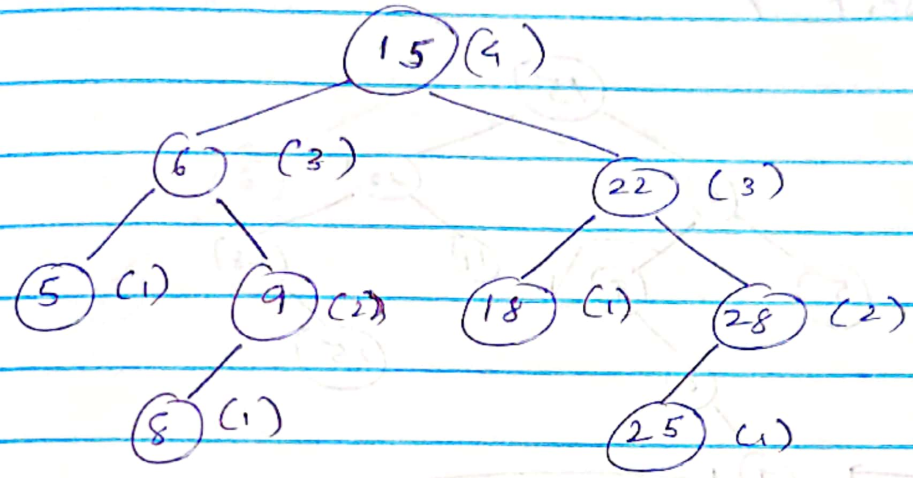
$h(59,0) = [59 + 0(10)] \bmod 11 = 4$ → occupied.

$h(59,1) = [59 + 1(10)] \bmod 11 = 3$ → occupied.

$h(59,2) = [59 + 2(10)] \bmod 11 = 2$ index

Q.2

a)



(height of leaf node is 1)

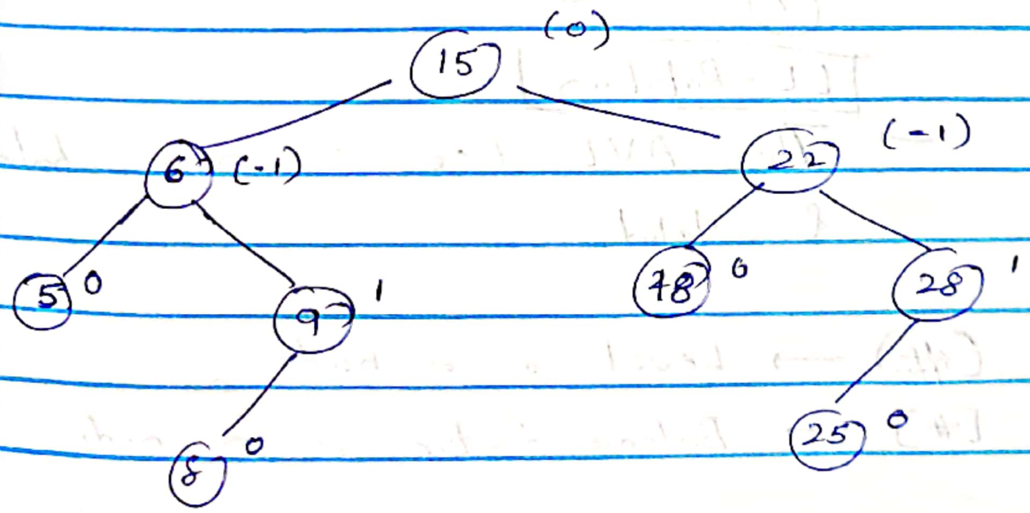
Height of :

8 = 1   5 = 1   22 = 3   15 = 4

25 = 1   6 = 3   28 = 2

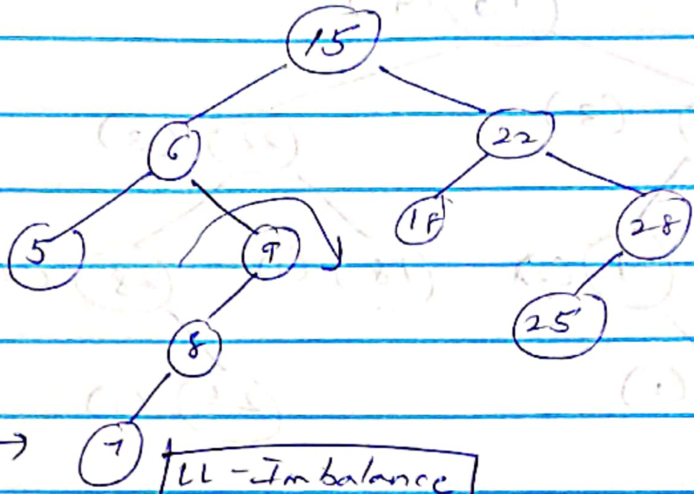
9 = 2   18 = 1   25 = 1

Redraw the tree with balance factor



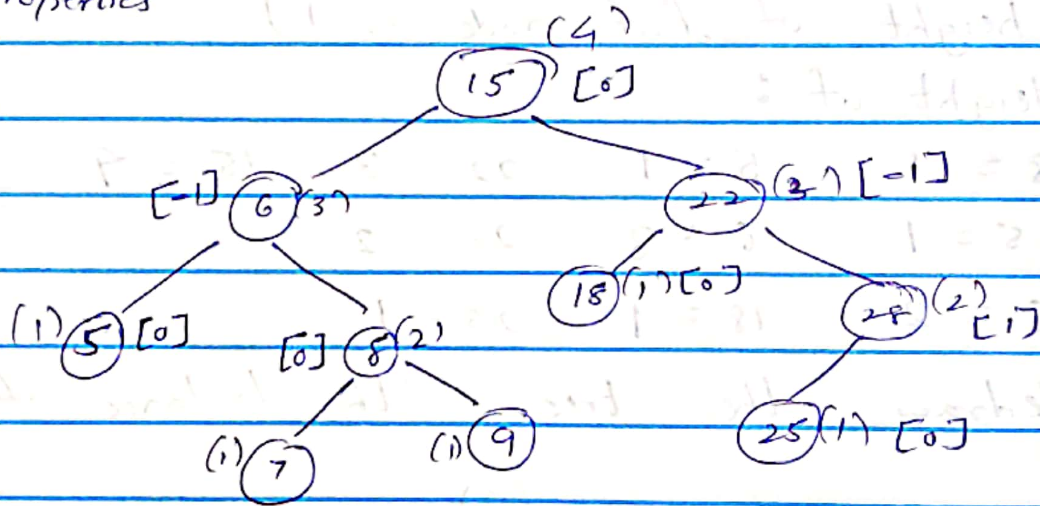
Since the balance factor of each node is less than or equal to 1, this is a Valid AVL tree.

b.) Insert 7



Inserted according to BST properties →

LL-Imbalance



LL-Rotation

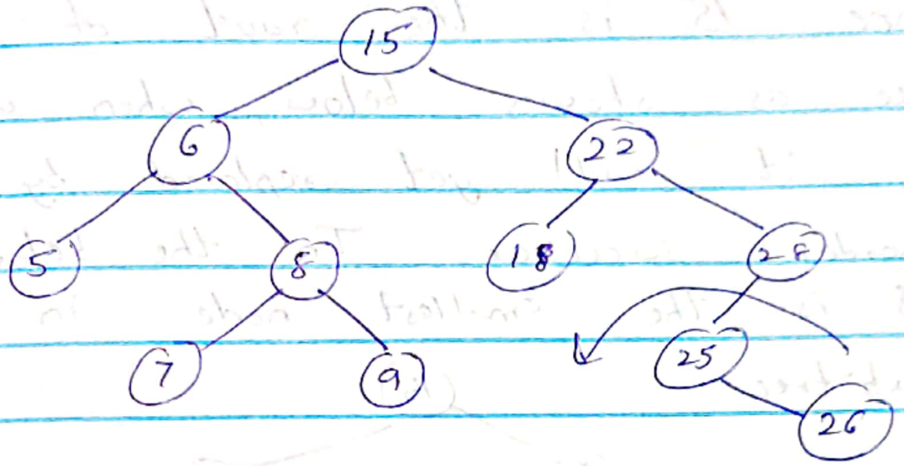
→ The AVL tree is now balanced & valid.

(#) → Level of a node

[#] → Balance factor of a node

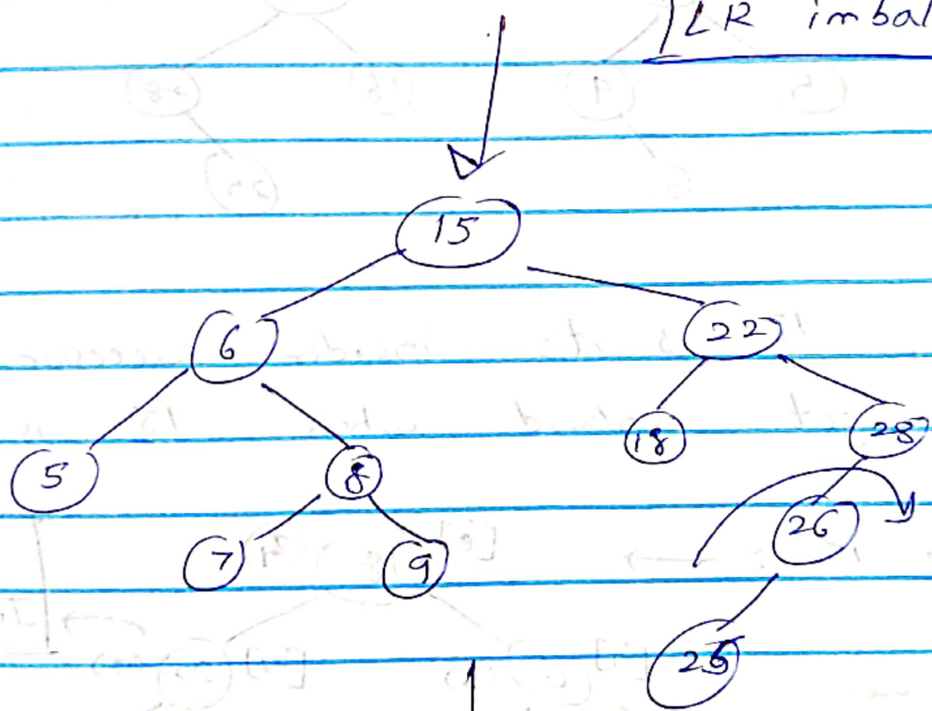
Scanned with OKEN Scanner

Insert 26

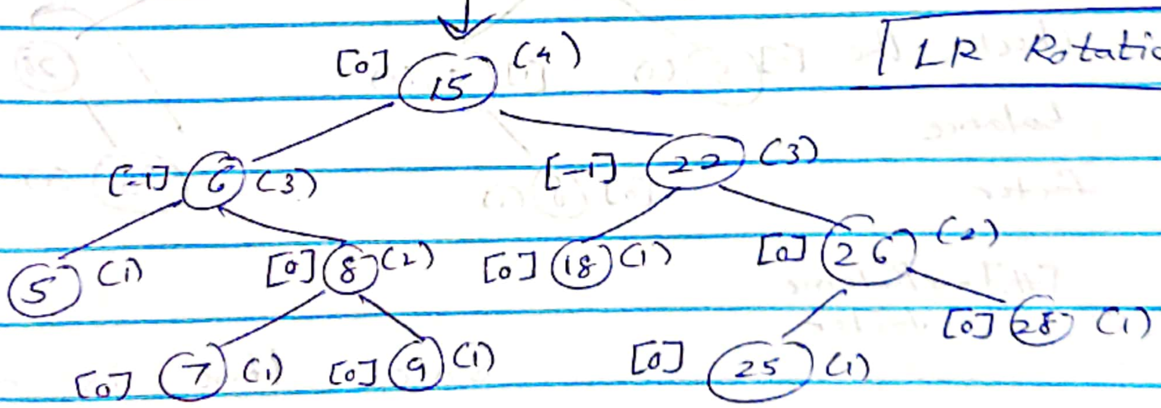


Inserted according to BST properties.

LR imbalance



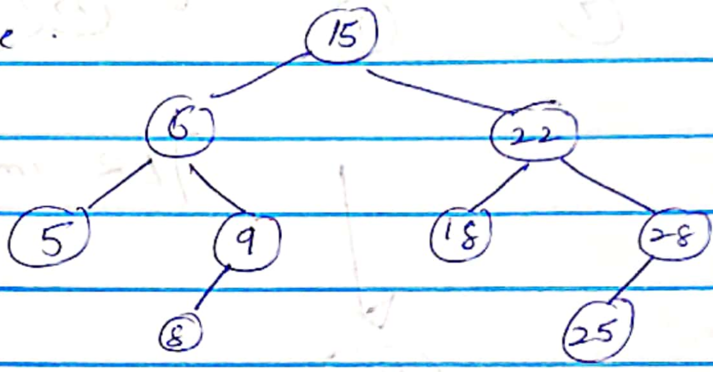
LR Rotation



The AVL tree is now balanced

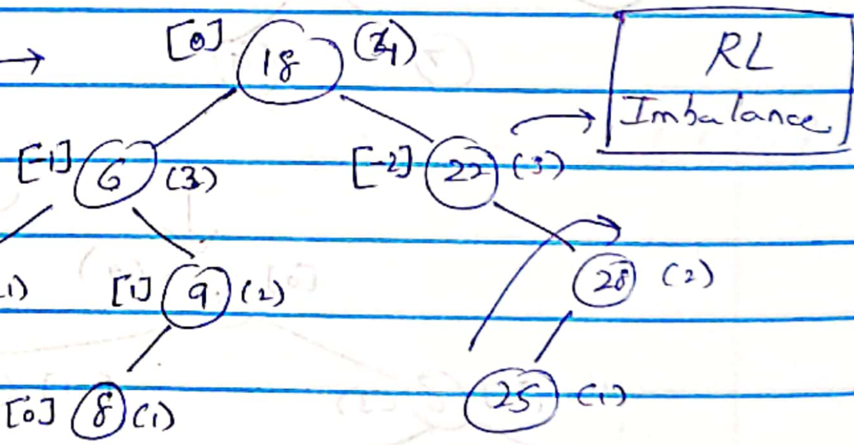
c.) Delete 15

Since 15 is the root of the AVL tree as shown below, when we delete 15, it will get replaced by its inorder successor. In the below tree, 18 is the smallest node in 15's right sub-tree.



Hence, 18 is it's inorder successor and will get replaced when 15 is deleted

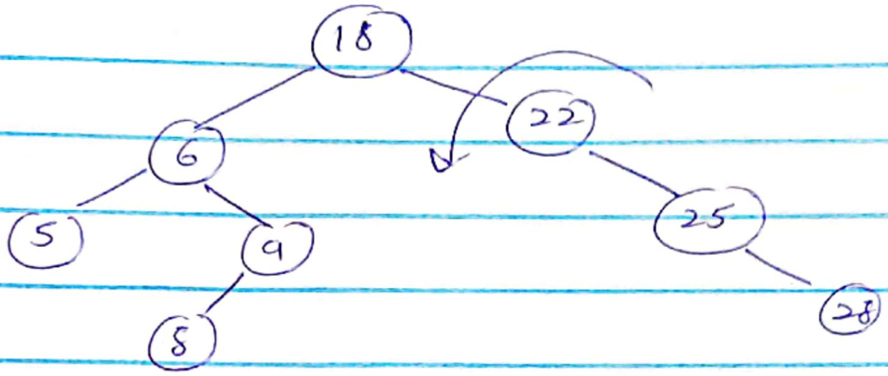
∴ Delete 15 : →

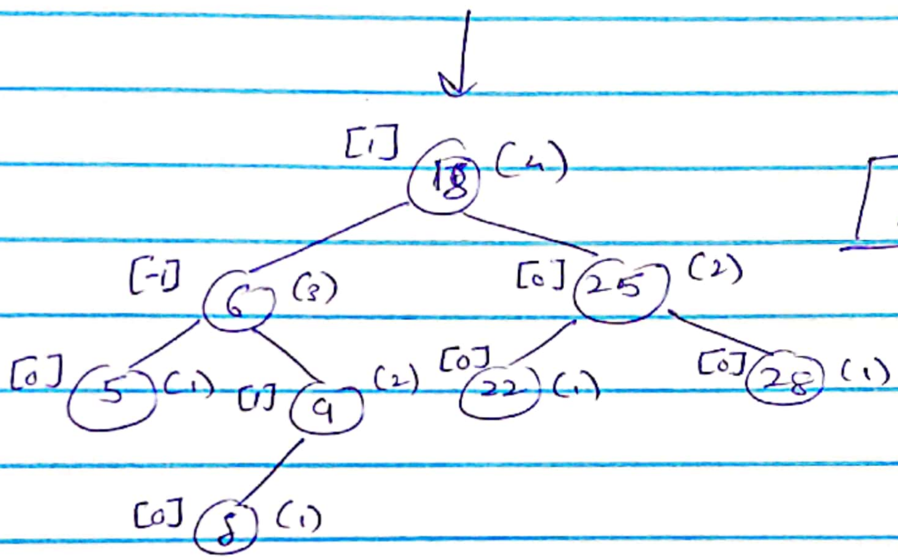


Now, we check the balance factor

[#] → Balance factor

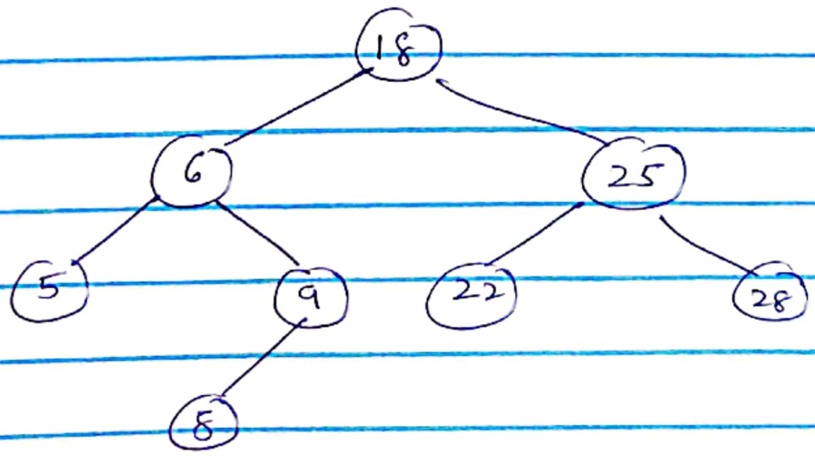
Since there is a imbalance at Not 22, we need to balance the tree





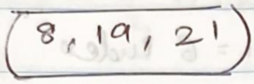
RL - Rotation

Now the tree is balanced and the final AVL is :

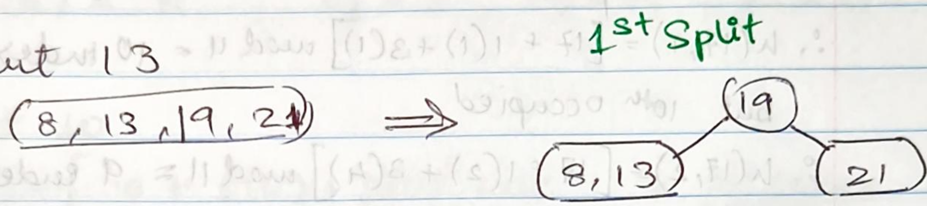


Q3. 8, 21, 19, 13, 5, 14, 10, 22, 24, 25, 15, 20, 16, 18, 3, 4, 26, 27, 6.

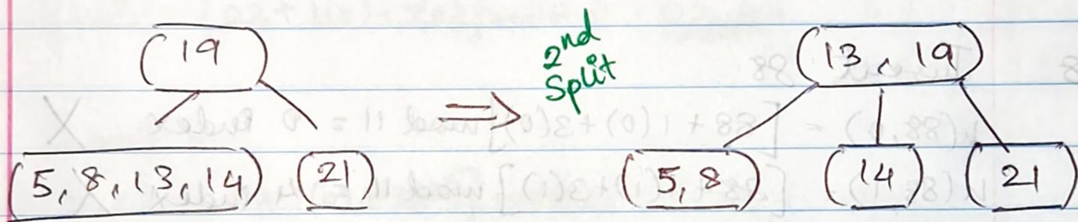
— Insert 8, 21, 19



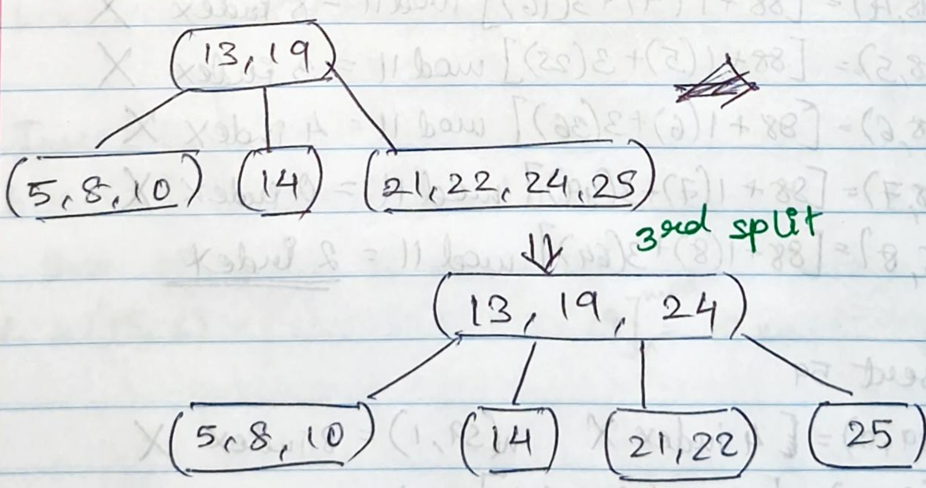
— Insert 13



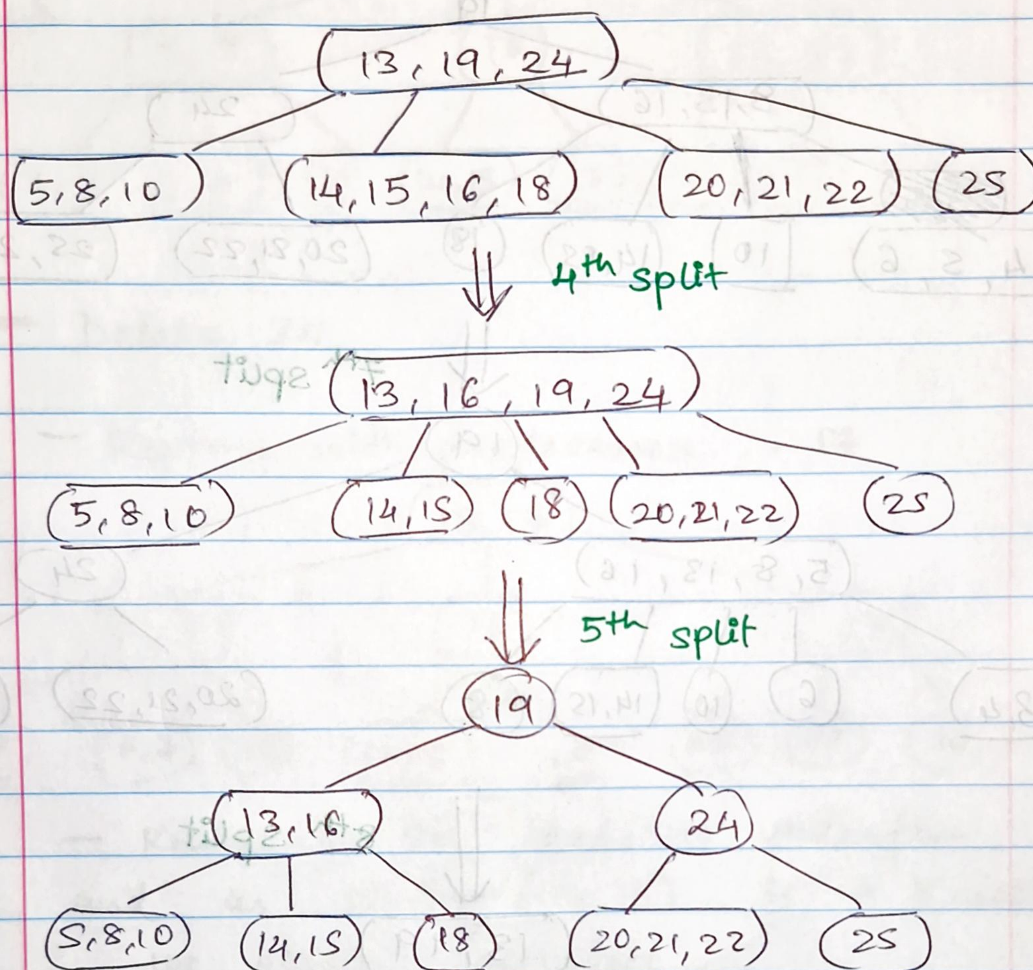
— Insert 5, 14



— Insert 10, 22, 24, 25

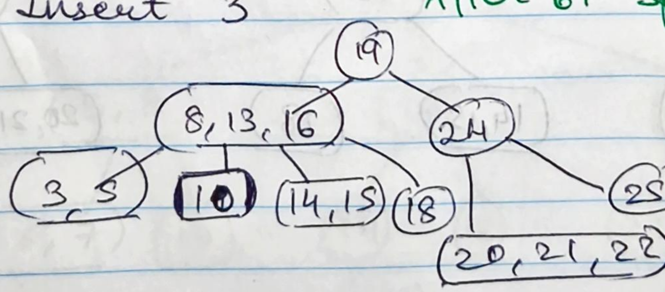


— Insert 15, 20, 16, 18

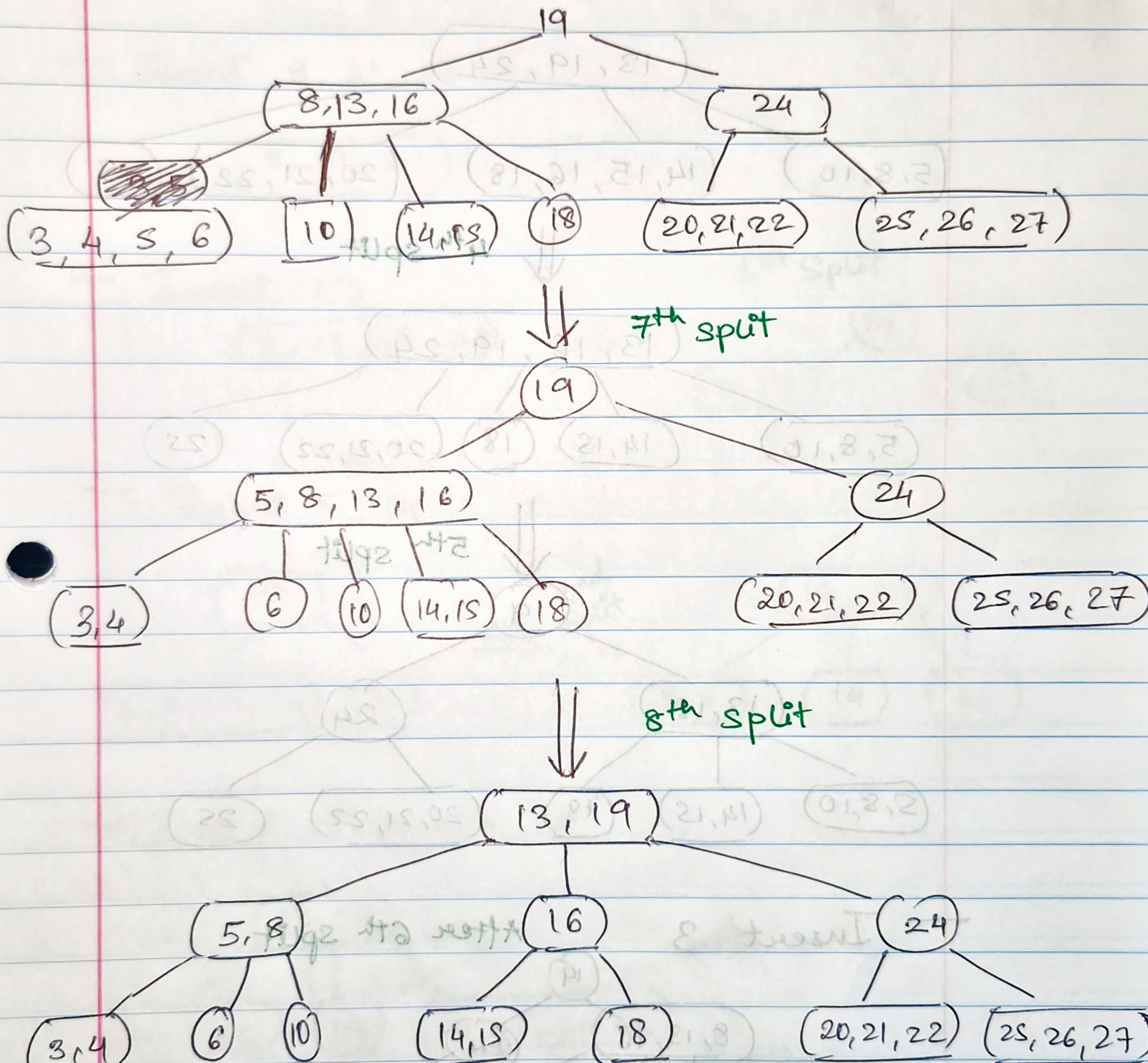


— Insert 3

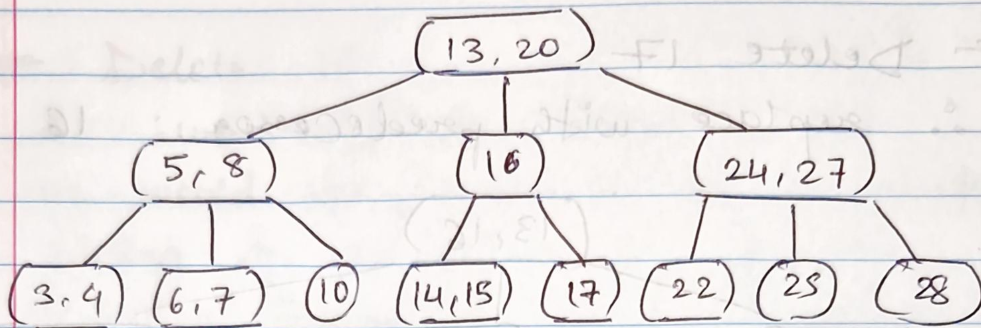
After 6th split



— Insert 4, 26, 27, 6

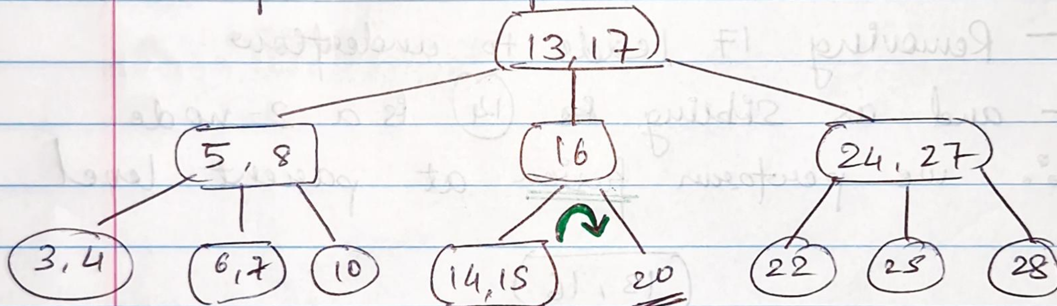


Q.4.

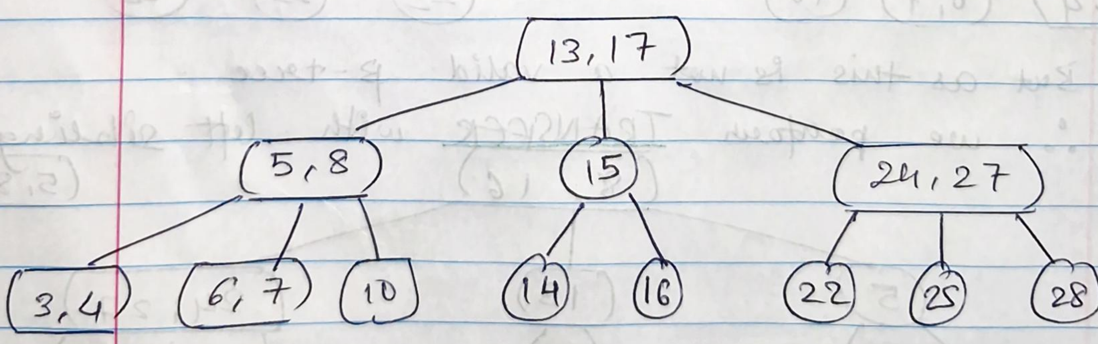


= Delete 20.

- Replace with predecessor ∴ 17

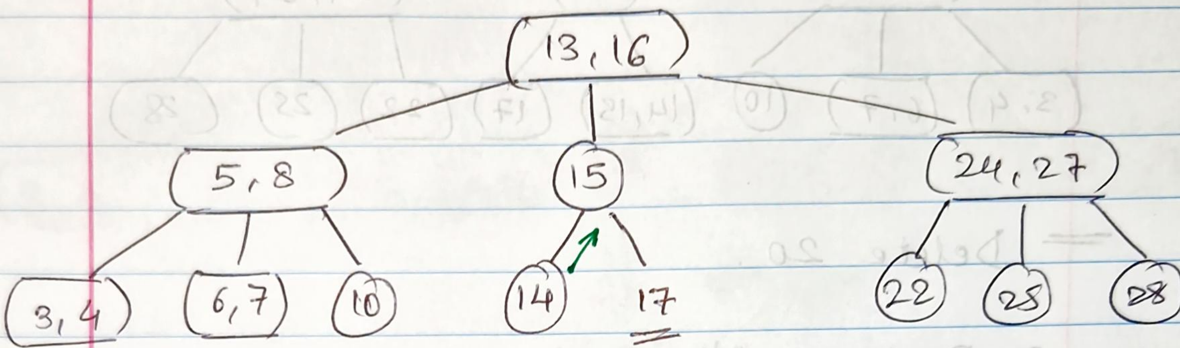


- Removing 20 leads to underflow and as sibling (14, 15) is a 3-node ∴ we perform transfer.



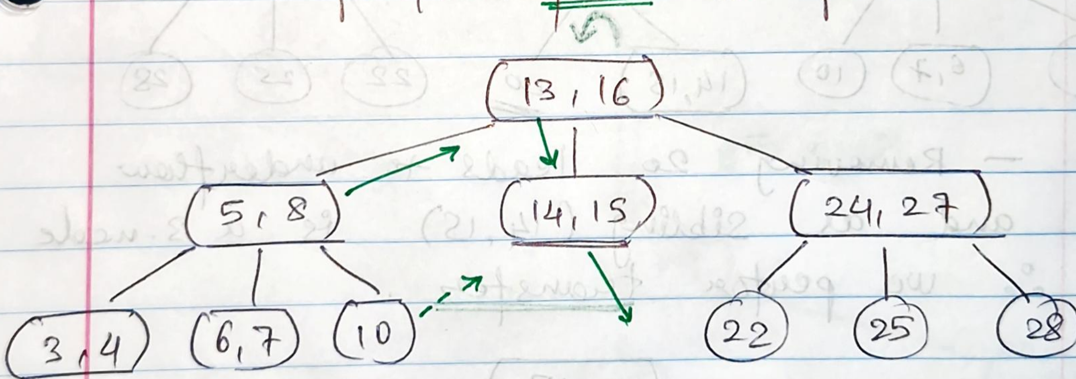
= Delete 17

∴ replace with predecessor : 16



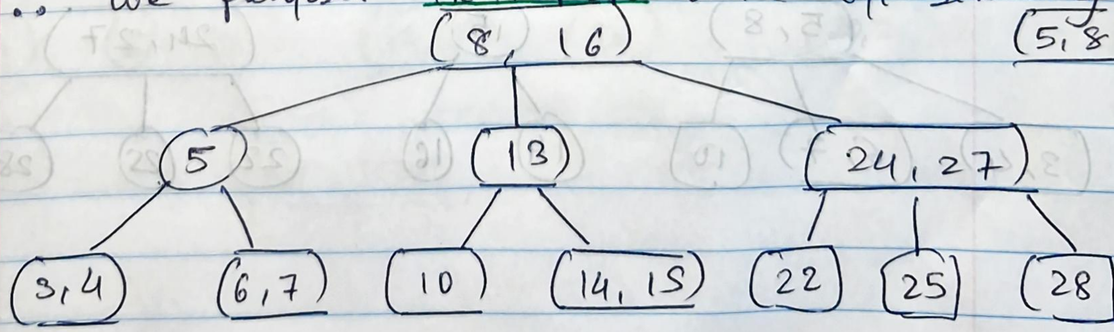
- Removing 17 leads to underflow.
- and as sibling ~~is~~ 14 is a 2-node

∴ we perform fusion at parent level



But as this is not a valid B-tree.

∴ we perform TRANSFER with left sibling. (5, 8)



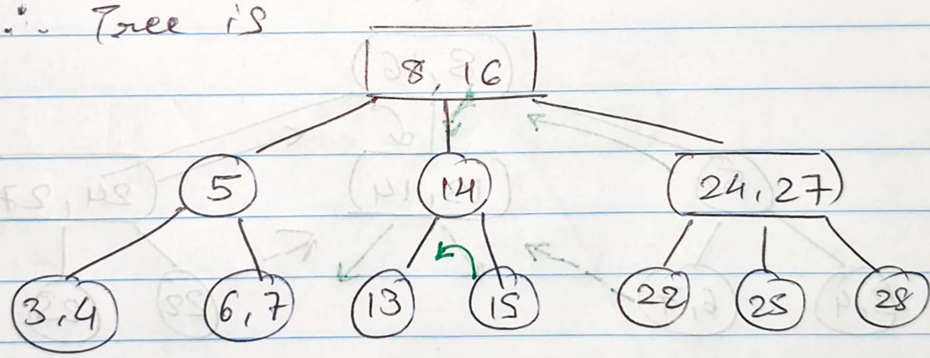
— Delete 10

As we will be deleting 10, we would not have a left immediate sibling & as given in question:

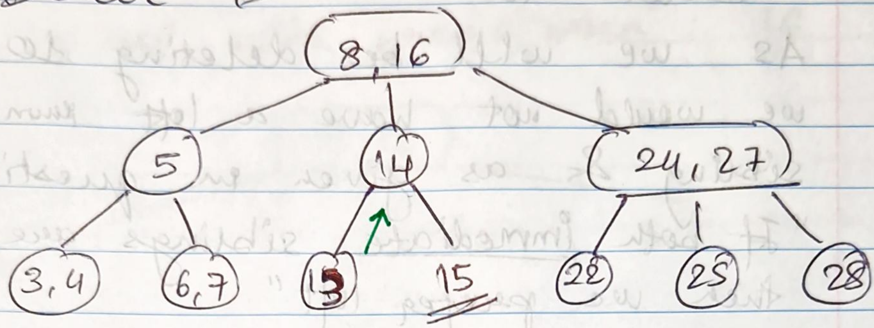
"If both immediate siblings are available then we prefer left"

So, as we only have 1 immediate sibling i.e the right sibling ∴ we will initiate a TRANSFER with (14, 15)

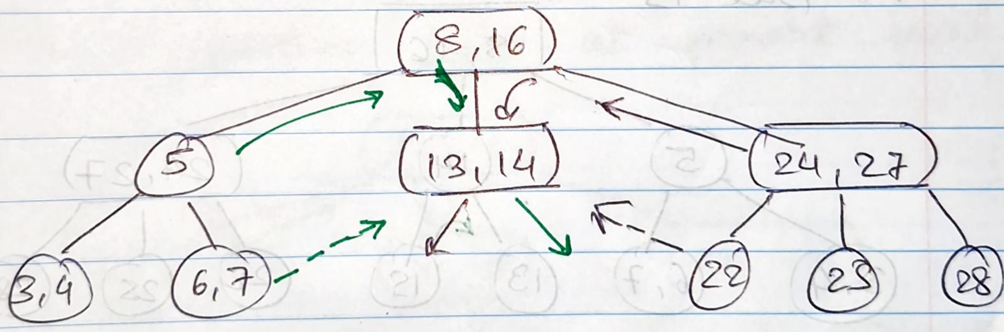
∴ Tree is



= Delete 15



If we delete 15 there will be an underflow and as immidiate left sibling (13) is a 2-node - we will perform FUSION



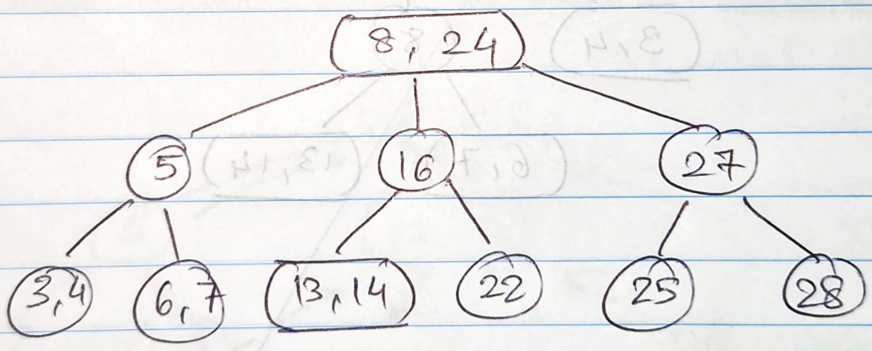
- But this is not a valid (2,4) tree as leaf node (13, 14) is not at same depth as other nodes.

- Also (13, 14) ~~DO NOT~~ ~~HAVE~~ HAS ~~any~~ immediate siblings (5) and (24, 27)

∴ we ~~prefer~~ have 2 options and the moves are marked in GREEN for left sibling & BLACK for right sibling

∴ we prefer right sibling ~~to~~ as it is a 3 node

∴ Tree is



This is the final tree

Hence solved.

Q.5.

a. A particular key is hashed into a slot with $(1/n)$ probability. Let there be k keys. The probability of these k keys are inserted into the slot in question is :

$$\left(\frac{1}{n}\right)^k \left(1-\frac{1}{n}\right)^{n-k}$$

Ways to choose the keys : $\binom{n}{k}$

$$\therefore Q_k = \left(\frac{1}{n}\right)^k \left(1-\frac{1}{n}\right)^{n-k} \binom{n}{k}$$

b. For $i = 1, 2, \ldots, n$, let X_i be a random variable denoting the no. of keys that hash to slot i. Let A_i be the event that $X_i = k$. that exactly k keys hash to slot i.

From (a) we have $P_r(A) = Q_k$

$$\begin{aligned}
P_k &= P_r(M = k) \\
&= P_r(\max(X_i : 1 \le i \le n) = k) \\
&= P_r(\exists\, X_i = k \ \& \ X_i \le k \text{ for } i = 1, 2, \ldots, n) \\
&\le P_r(\exists\, i \text{ s.t. } X_i = k) \\
&= P_r(A_1 \cup A_2 \cup \ldots \cup A_n) \\
&\le P_r(A_1) + P_r(A_2) + \ldots + P_r(A_n) = \boxed{n \cdot Q_k}
\end{aligned}$$

c. To show $Q_k < \frac{1}{k!}$

We need to show :

$1 - \frac{1}{n} < 1$ & $n-k \geq 0$

$\therefore \left(1 - \frac{1}{n}\right)^{n-k} \leq 1$

and

$\frac{n!}{(n-k)!} = n.(n-1)(n-2) \cdots (n-k+1) < n^k.$

By this and using $k! > \left(\frac{k}{e}\right)^k$ we have

$Q_k = \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \frac{n!}{k!(n-k)!}$

$\leq \frac{n!}{n^k k!(n-k)!}$ $\quad ((1 - 1/n)^{n-k} < 1)$

$< 1/k!$ $\quad (n!/(n-k)! < n^k)$

$< e^k/k^k$ $\quad (k! > (k/e)^k)$

$<$ $\boxed{Q_k < \frac{1}{k!}}$

e. The expectation of M is

$$E[M] = \sum_{k=0}^{n} k \cdot Pr(M=k)$$

$$= \sum_{k=0}^{k_0} k \cdot Pr(M=k) + \sum_{k=k_0+1}^{n} k \cdot Pr(M=k)$$

$$\leq \sum_{k=0}^{k_0} k_0 \cdot Pr(M=k) + \sum_{k=k_0+1}^{n} n \cdot Pr(M=k)$$

$$\leq k_0 \sum_{k=0}^{k_0} Pr(M=k) + n \sum_{k=k_0+1}^{n} Pr(M=k)$$

$$= k_0 \cdot Pr(M \leq k_0) + n \cdot Pr(M > k_0)$$

Since $k_0 = \dfrac{c \log n}{\log(\log n)}$

Now, $Pr(M > k_0) = \sum_{k=k_0+1}^{n} Pr(M=k)$

$$= \sum_{k=k_0+1}^{n} P_k$$

$$< \sum_{k=k_0+1}^{n} 1/n^2$$

$$< n \cdot (1/n^2)$$

$$= 1/n$$

Scanned with OKEN Scanner

Substituting the values of $Pr(M > k_0)$ in $E[M]$ we get,

$$E[M] \le k_0 \cdot 1 + n \cdot (1/n)$$
$$= k_0 + 1$$
$$= \boxed{O\left(\frac{\log n}{\log(\log n)}\right)}$$

Thus Proved

6) i) Create an empty hash set H.
ii) Iterate through array of n integers:
$\forall x \rightarrow$ insert x into H
iii) Iterate through array again:
$\forall x \rightarrow$ check if $x+1$ or $x-1$ exists in H
If $x+1$ exists in H, report pair $(x, x+1)$
If $x-1$ exists in H, report pair $(x-1, x)$
iv) Return all reported pairs

Time complexity -
$O(n)$ expected time due to hashing operations

Space complexity -
$O(n)$ worst case space for storing n elements in the hash set