

Assignment 6

Anusha Kothapally

1)
(a)

Given order, $m=4$

B+ tree: Key values: 2, 3, 5, 7, 11, 17, 19, 23, 29, 31,
9, 10, 8

Order = m , $\lceil \frac{m}{2} \rceil = 2$

Step 1: insert: 2, 3, 5, 7

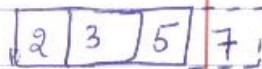


figure 2: insert: 11, 17

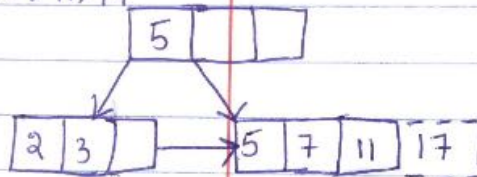


figure 3: insert: 19, 23

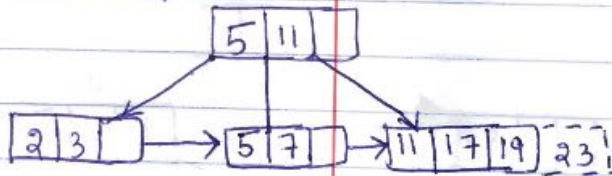
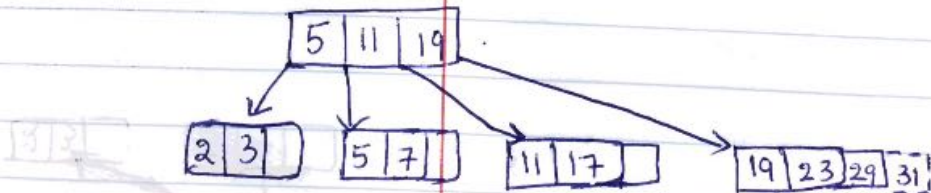


figure 4: insert: 29, 31



1)
(a)

Figure 5: (Intermediary figure)

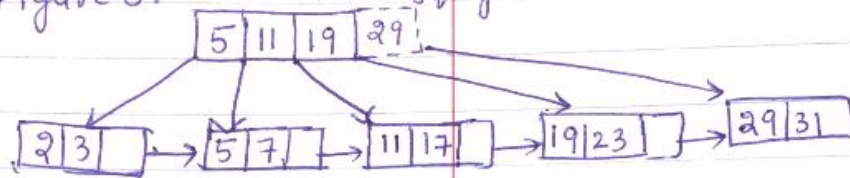


Figure 6: Insert: 9, 10

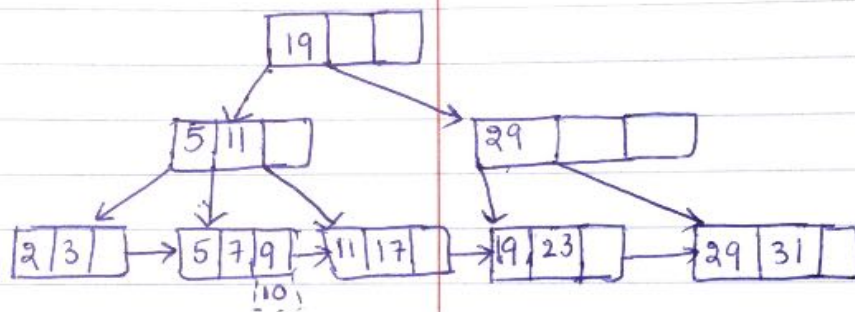
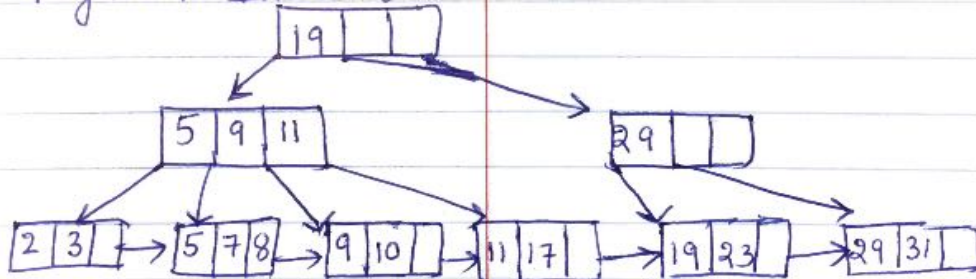


Figure 7: Insert: 8



1)
(a)

Delete 23, violates the condition, therefore underflow node-containing 19 is merged with its sibling node, now the root is updated to 19. Again, there is violation at internal parent node 19, did not have sufficient number of child nodes; therefore, it is merged with its sibling and reordered.

Figure 8:

Delete 23. It violates condition that leaf node should have values between $\lceil \frac{n-1}{2} \rceil$ and $n-1$

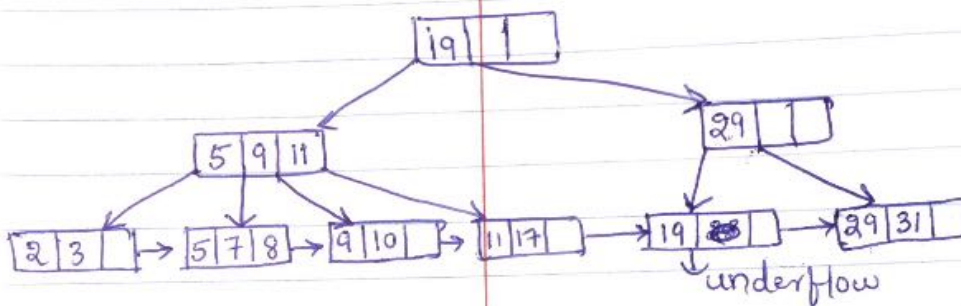
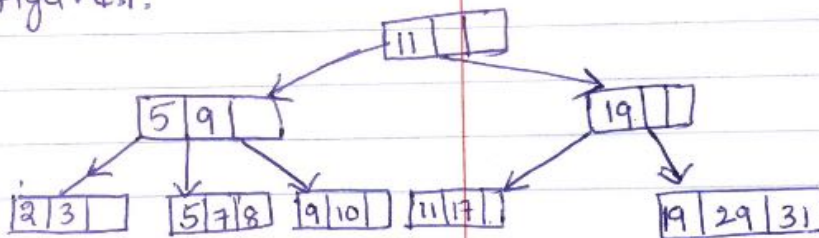


Figure 9:

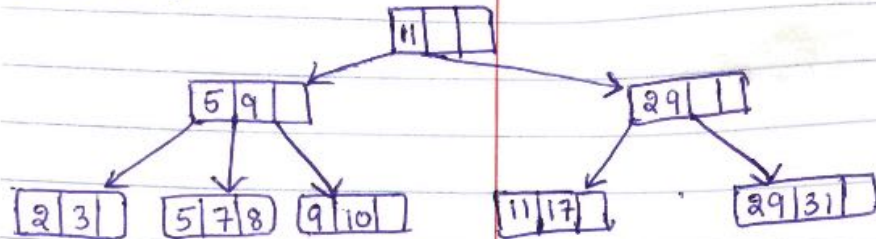


Here, we have joined with 19 with [29 | 31].
∴ therefore, it became [19 | 29 | 31] and root becomes 19, again this violated no. of children's condition therefore, reordered.

1)
(a)

Delete key-value 19, can be carried directly.

Figure 10:
Delete 19:



1)
(b)

⑤ order=6 Key values: 2, 3, 5, 7, 11, 17, 19, 23, ²⁹31, 9, 10, 8

insert: 2, 3, 5, 7, 11, 17

figure 1:

2	3	5	7	11	7
---	---	---	---	----	---

 - overflow.

figure 2: insert: 19, 23, 29

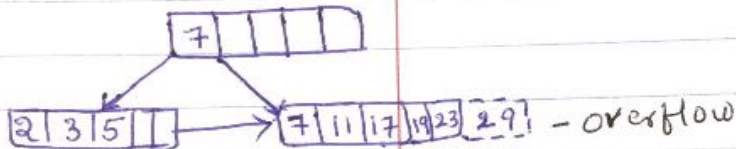


figure 3: insert: 31

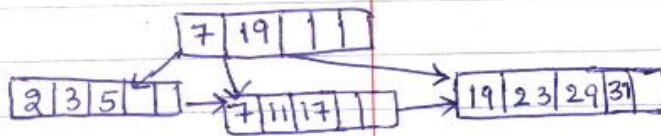


figure 4: insert: 9

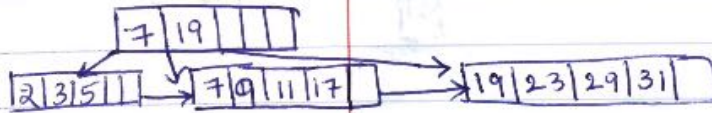


figure 5: insert: 10, 8

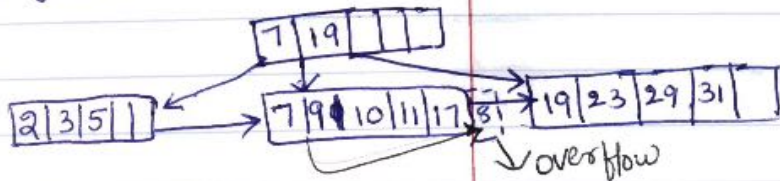


fig 6:

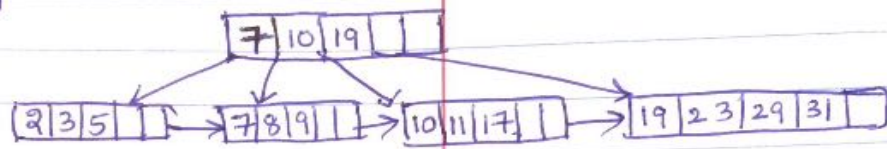


fig 7: Delete '23'.

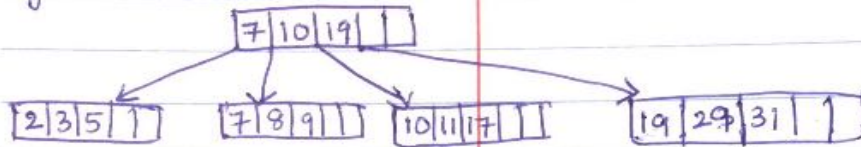
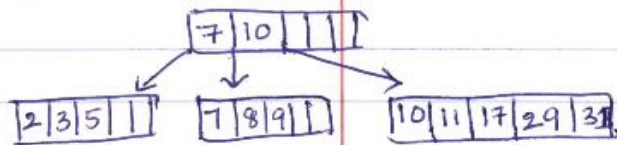


fig 8: Delete '19' - will cause underflow in the [19, 29, 31] in fig 7. Therefore, we could join [10, 11, 17] with [29, 31] after deleting '19'.

fig 9:



1)
(c)

© With order=6, there are fewer node splits and merges compare to order=4. But while searching for a element at node, since it contains more elements with order=6 compare to order=4, we may need more comparisons within node, that is more cpu time. No. of levels in B+tree (if it is more) requires I/O access. I/O is costlier compare. to cpu time.

2)

② Extendable hashing:

hash prefix:

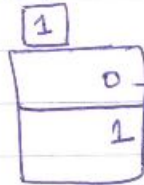


0		
A-305	Round Hill	350
A-222	Red wood	700

bucket 1

insert: perry ridge insertion at bucket 1 will cause overflow ~~in~~ therefore split into two buckets.

hash prefix:



1		
Red wood	A-222	700

bucket 1

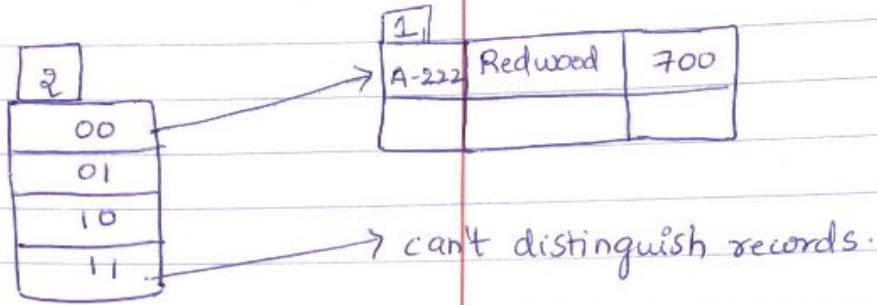
1		
A-305	Round Hill	350
A-218	Perry ridge	700

bucket 2

insert: again perry ridge - A-201, should go to bucket 2, since its starting bit is '1' but there is a overflow at bucket 2. Therefore split into two buckets.

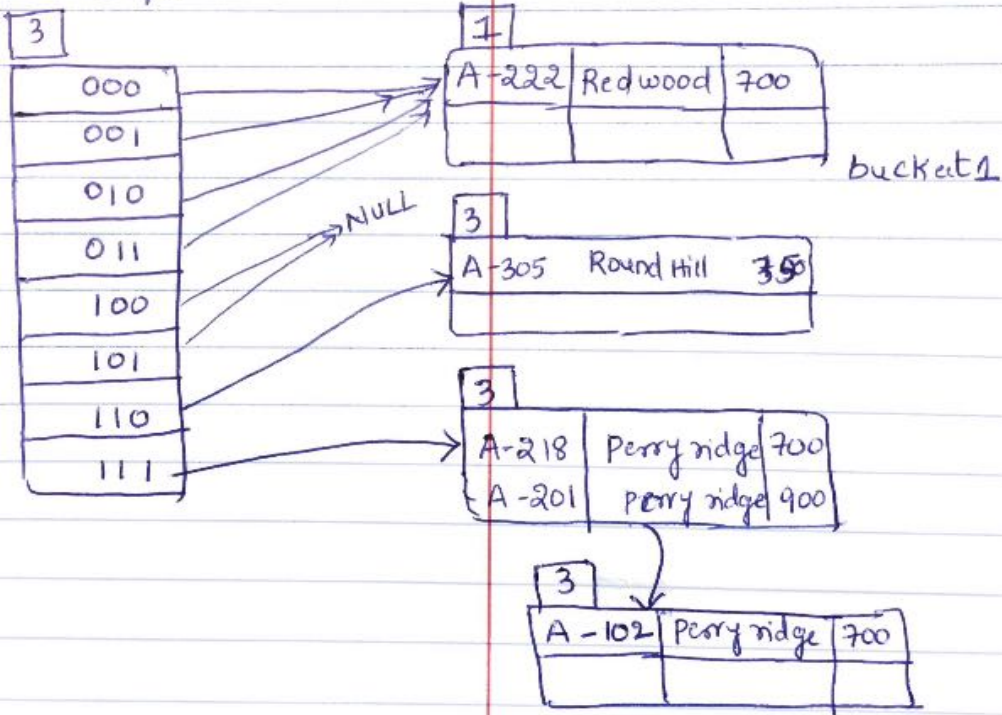
If we use hash prefix we can't actually distinguish Perry ridge and Round Hill, since the 2 MSBits in both of them is same, we need to consider additional bit to distinguish them.

That is, with hash prefix = 2



Therefore, insert: Perry ridge A-201, Perry ridge A-102

hash prefix



Insert: A-215 Mianus, Downtown A-110, Downtown A-101

