



CS 443

# LAB 5

Malialosa Taupule

2 May 2017



**Question 1:**

A PARTS file with Part# as hash key includes records with the following Part# values: 2369, 3760, 4692, 4871, 5659, 1821, 1074, 7115, 1620, 2428, 3943, 4750, 6975, 4981, 9208. The file uses 8 buckets, numbered 0 to 7. Each bucket is one disk block and holds two records. Load these records into the file in the given order using the hash function  $h(K)=K \bmod 8$ . Calculate the average number of block accesses for a random retrieval on Part#.

K	$h(K) = K \% 8$	Bucket #	Pointer
2369	$2369 \% 8$	1	NULL
3760	$3760 \% 8$	0	NULL
4692	$4692 \% 8$	4	NULL
4871	$4871 \% 8$	7	NULL
5659	$5659 \% 8$	3	NULL
1821	$1821 \% 8$	5	NULL
1074	$1074 \% 8$	2	NULL
7115	$7115 \% 8$	3	NULL
1620	$1620 \% 8$	4	NULL
2428	$2428 \% 8$	4	
3943	$3943 \% 8$	7	NULL
4750	$4750 \% 8$	6	NULL
6975	$6975 \% 8$	7	
4981	$4981 \% 8$	5	NULL
9208	$9208 \% 8$	0	NULL

**Overflow area**

K	Bucket #	Pointer
2428	4	NULL
6975	7	NULL

$$\text{Avg. number of block accesses} = \sum \text{Event} \times P(\text{Event})$$

Event = # of accesses = 1 or 2 (depending if K is put into overflow area)

$$= \left(1 \times \frac{13}{15}\right) + \left(2 \times \frac{2}{15}\right) = .867 + .267 = 1.134 \text{ block accesses}$$

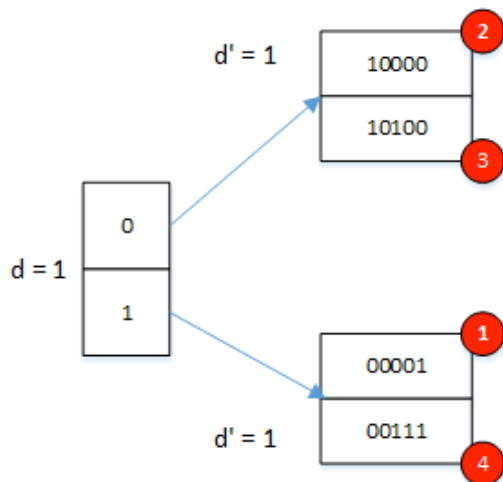
**Question 2:**

Consider the following records of question 1:

2369, 3760, 4692, 4871, 5659, 1821, 1074, 7151, 1620

Load the records into expandable hash files based on extendible hashing. You can put two records per block. Show the directory at each step, and the global and local depths. Use the hash function  $h(k) = K \bmod 32$  to find the bucket number.

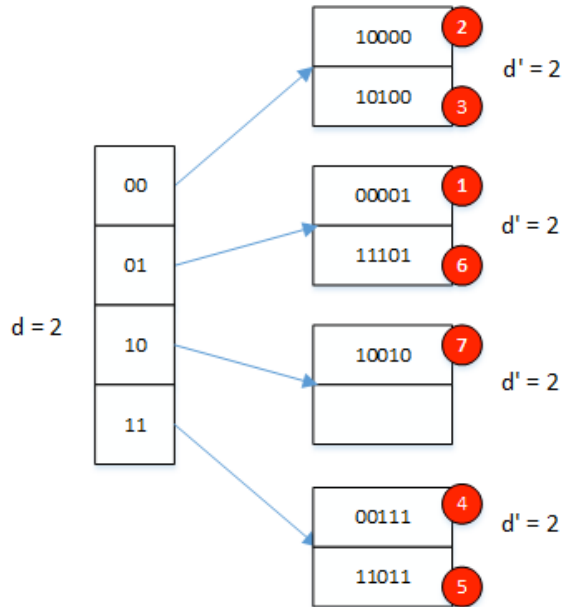
Record	K	$h(K)$ Bucket #	Binary $h(K)$
1	2369	1	00001
2	3760	16	10000
3	4692	20	10100
4	4871	7	00111
5	5659	27	11011
6	1821	29	11101
7	1074	18	10010
8	7151	15	01111
9	1620	20	10100

**Step 1) Create a directory with 1 bit and 2 buckets.****Insertion order:**

(indicated by red dots)

1. Insert 00001  $\rightarrow$  Bucket 1
2. Insert 10000  $\rightarrow$  Bucket 0
3. Insert 10100  $\rightarrow$  Bucket 0
4. Insert 00111  $\rightarrow$  Bucket 1

**Step 2) To insert record 5, increase directory size to 2.**

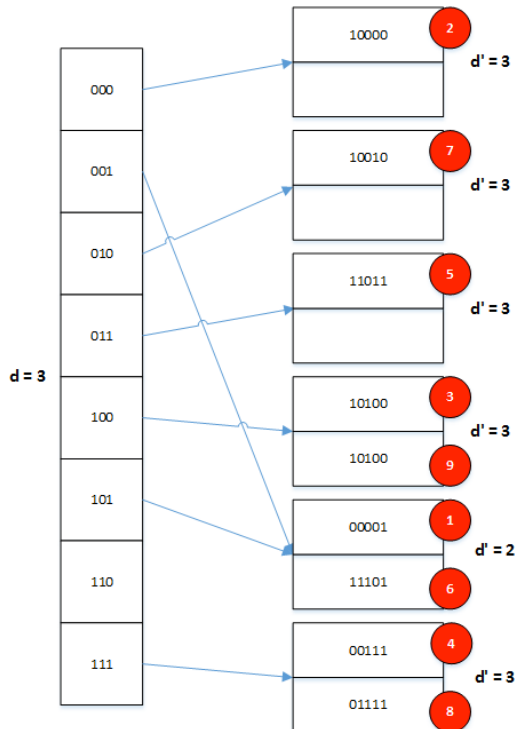


**Insertion order:**

(indicated by red dots)

1. Move 00001 → Bucket 01
2. Move 10000 → Bucket 00
3. Move 10100 → Bucket 00
4. Move 00111 → Bucket 11
5. Insert 11011 → Bucket 11
6. Insert 11101 → Bucket 01
7. Insert 10010 → Bucket 10

**Step 3) To insert record 8, increase directory size to 3.**



**Insertion order:**

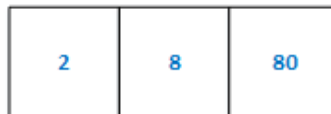
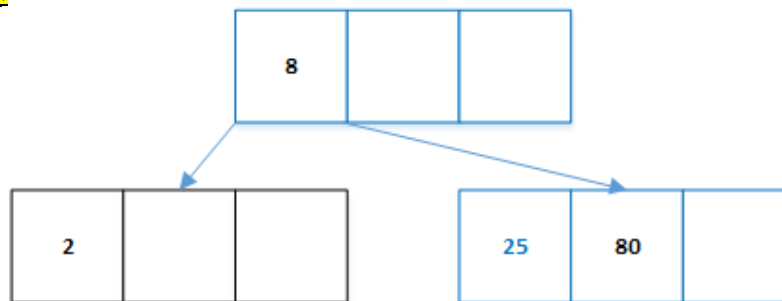
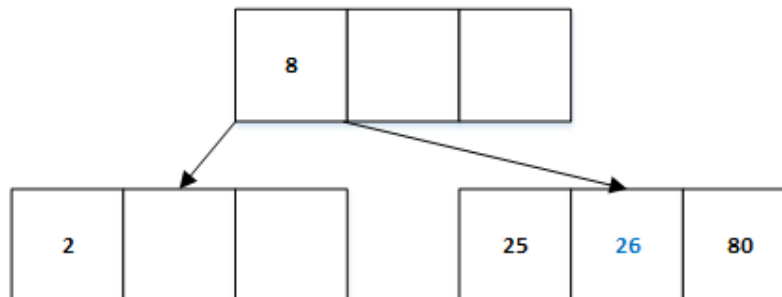
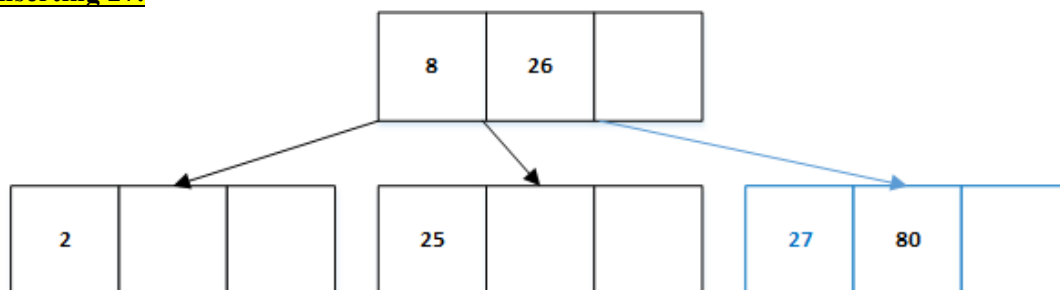
(indicated by red dots)

1. Move 00001 → Bucket 01
2. Move 10000 → Bucket 000
3. Move 10100 → Bucket 100
4. Move 00111 → Bucket 111
5. Move 11011 → Bucket 011
6. Move 11101 → Bucket 01
7. Move 10010 → Bucket 010
8. Insert 01111 → Bucket 111
9. Insert 10100 → Bucket 100

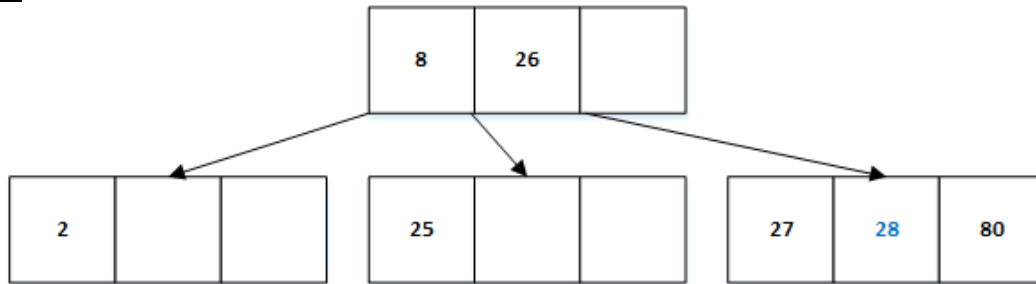
**Question 3:**

Insert the following into B-tree of order 4. Show your work step by step with proper illustration of pointers as shown in pages 20-29 in multi-way trees lecture

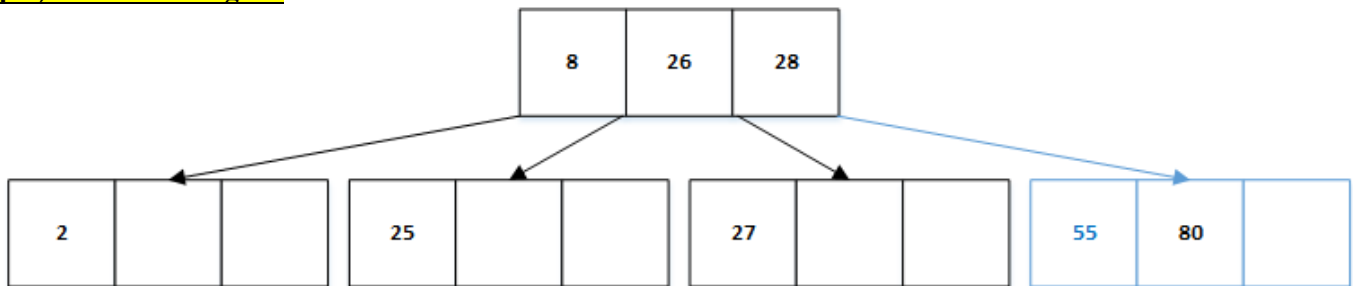
8, 2, 80, 25, 26, 27, 28, 55, 71, 15, 51, 20, 21, 22, 23, 24, 63, 90, 35

**Step 1) Insert 8, 2, 80.****Step 2) After inserting 25.****Step 3) Insert 26.****Step 4) After inserting 27.**

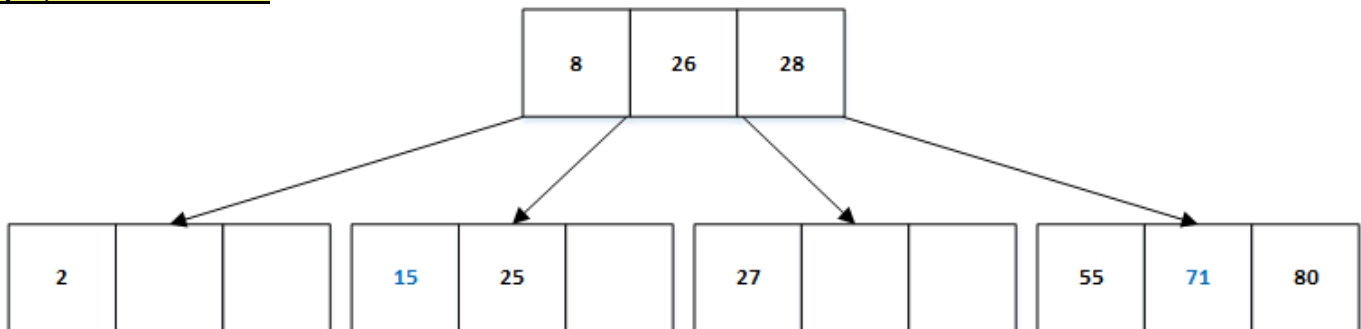
**Step 5) Insert 28.**



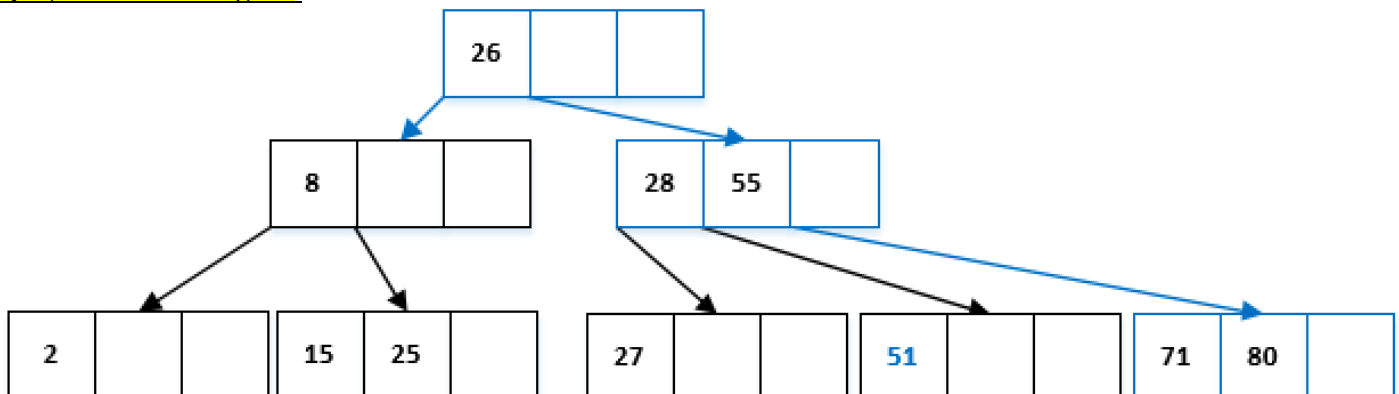
**Step 6) After inserting 55.**



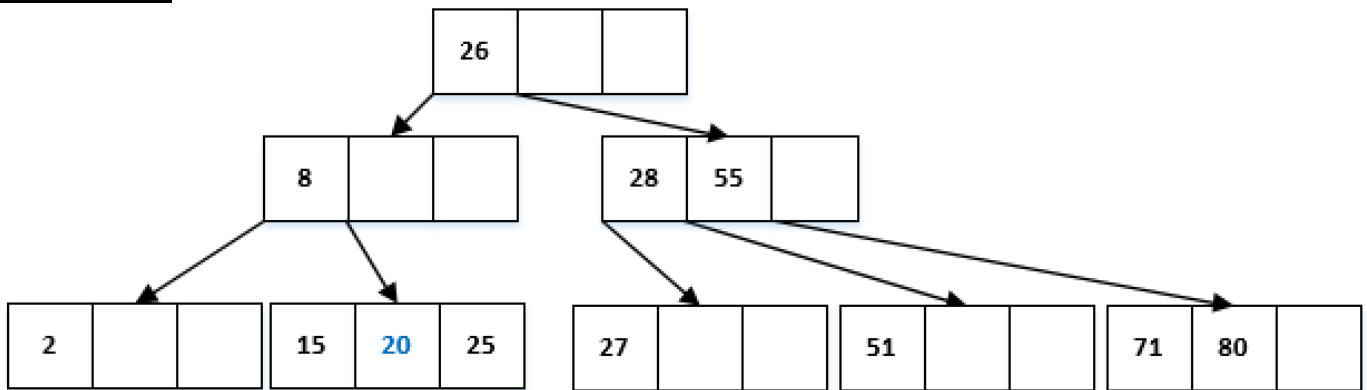
**Step 7) Insert 71 and 15.**



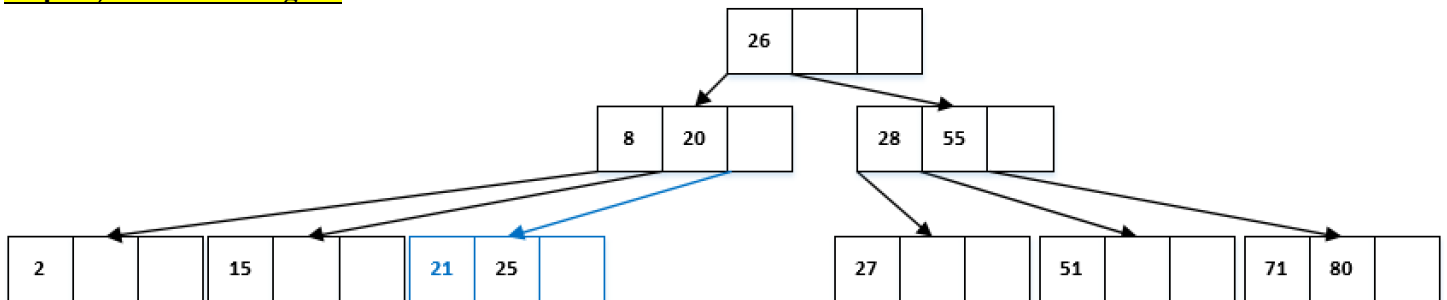
**Step 8) After inserting 51.**



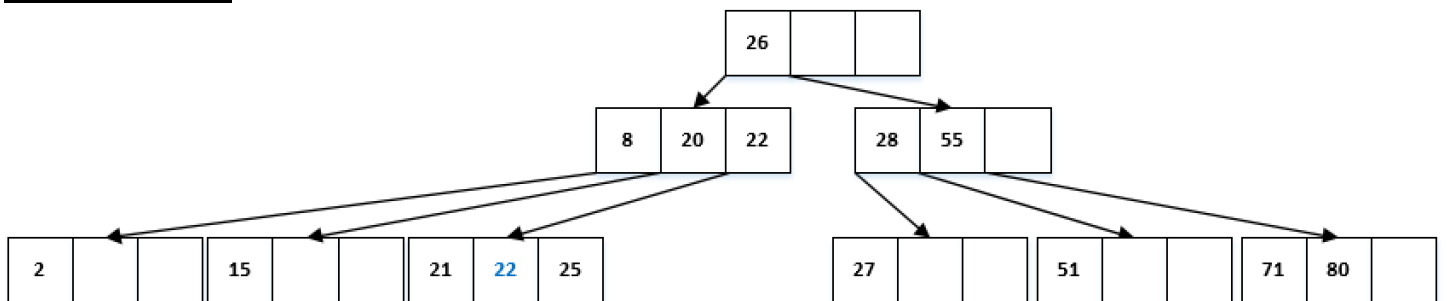
**Step 9) Insert 20.**



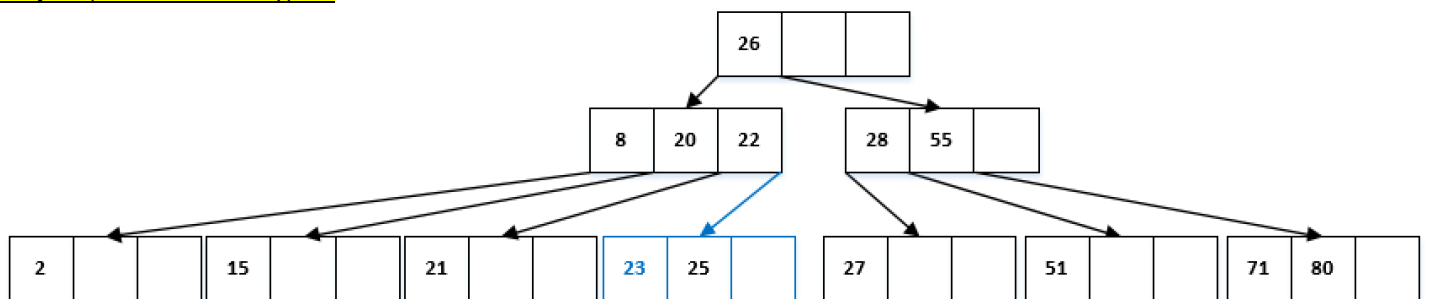
**Step 10) After inserting 21.**



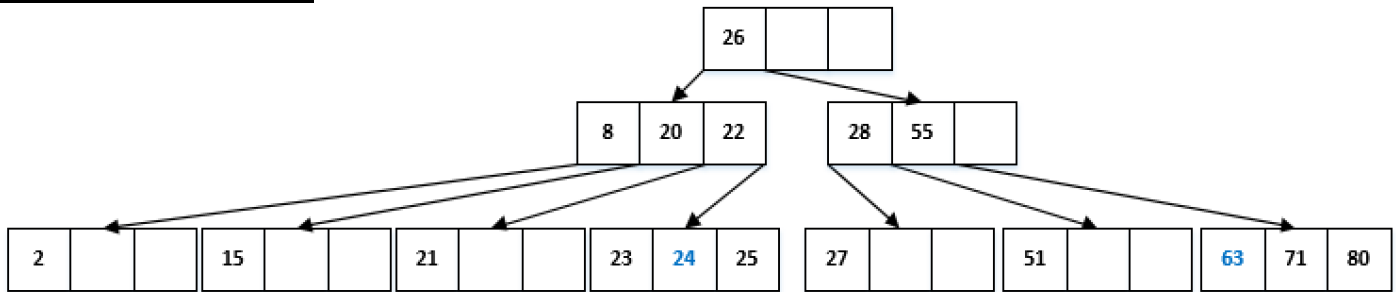
**Step 11) Insert 22.**



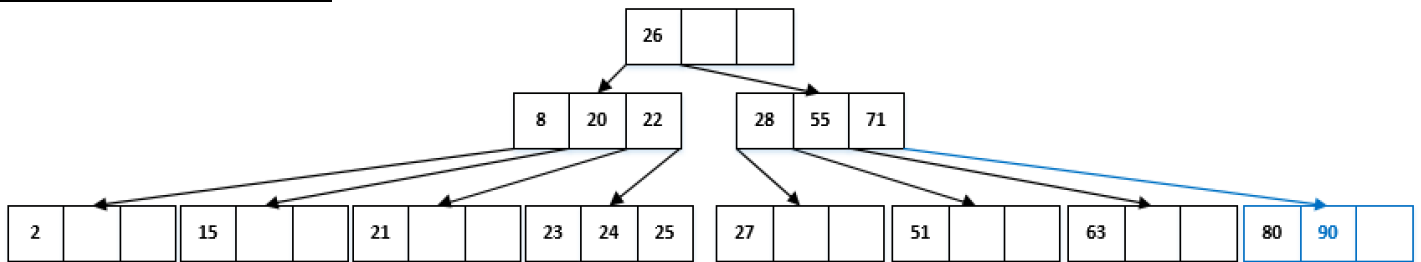
**Step 12) After inserting 23.**



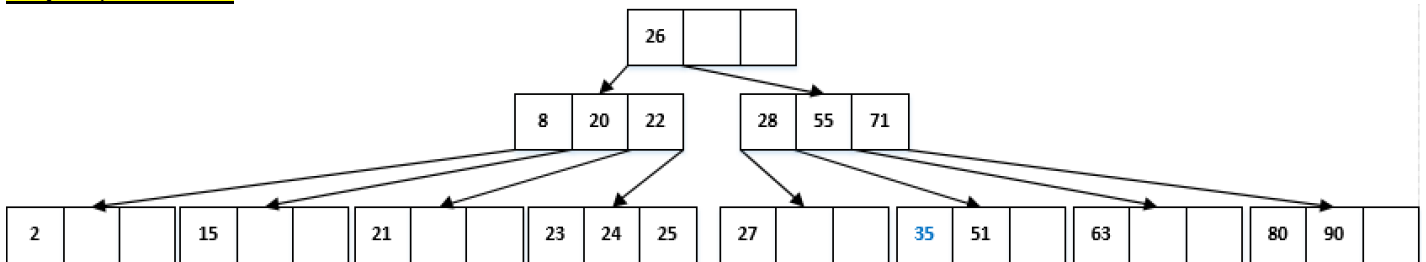
**Step 13) Insert 24 and 63.**



**Step 14) After inserting 90.**



**Step 15) Insert 35.**

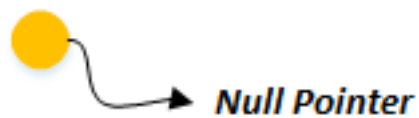
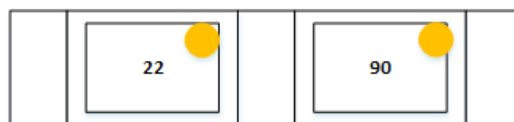
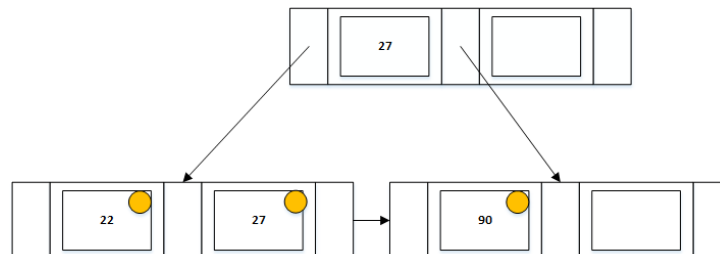
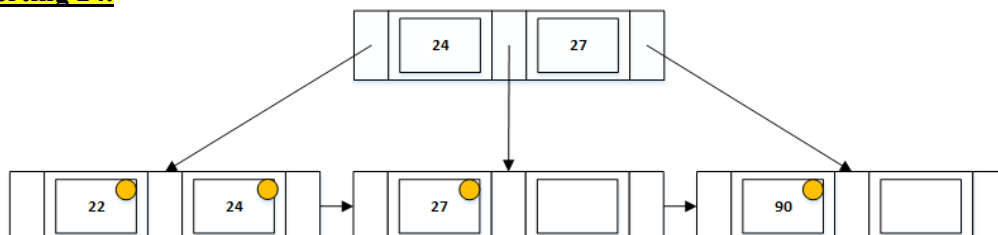




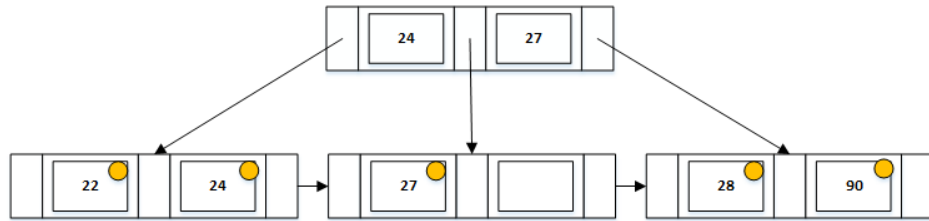
**Question 4:**

Insert the following into B<sup>+</sup> tree of order 3. Show your work step by step with proper illustration of pointers as shown in pages 47-54 in multi-way trees lecture

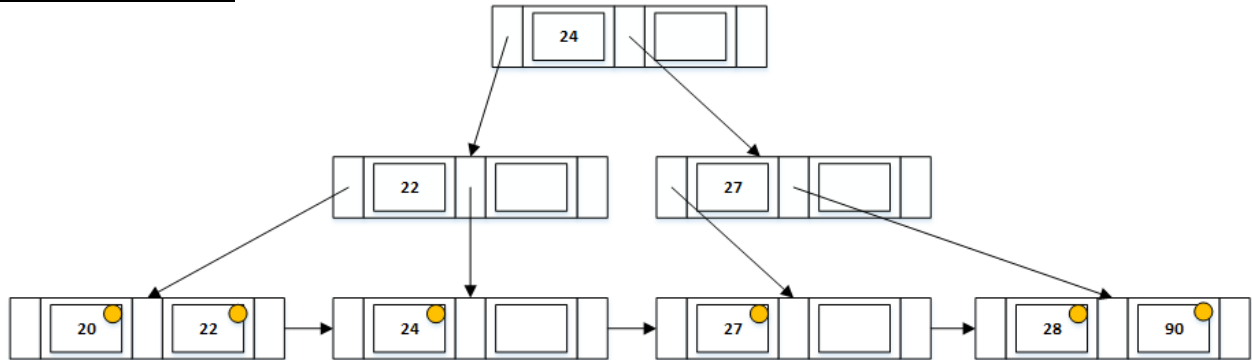
90, 22, 27, 24, 28, 20, 51, 63, 8, 80, 15, 71, 35, 55

**Start.****Step 1) Insert 90 and 22.****Step 2) After inserting 27.****Step 3) After inserting 24.**

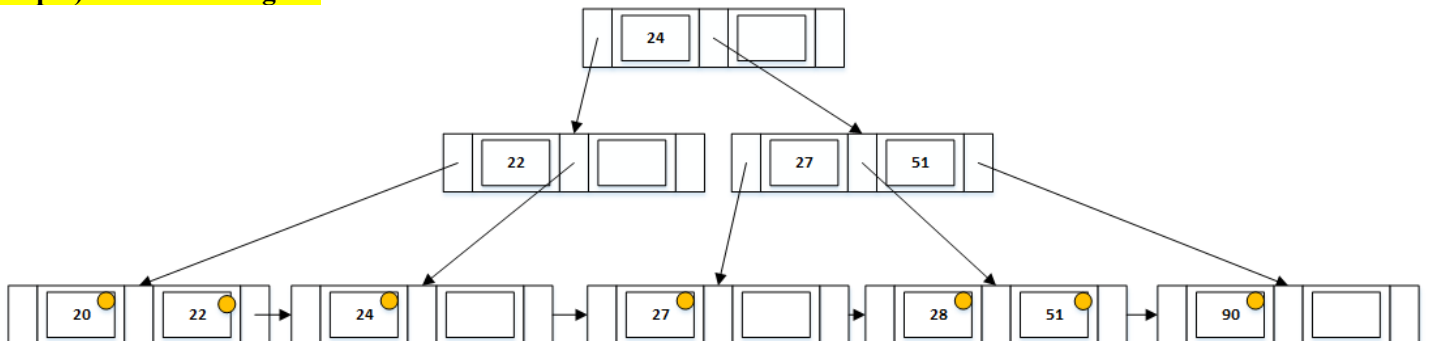
**Step 4) Insert 28.**



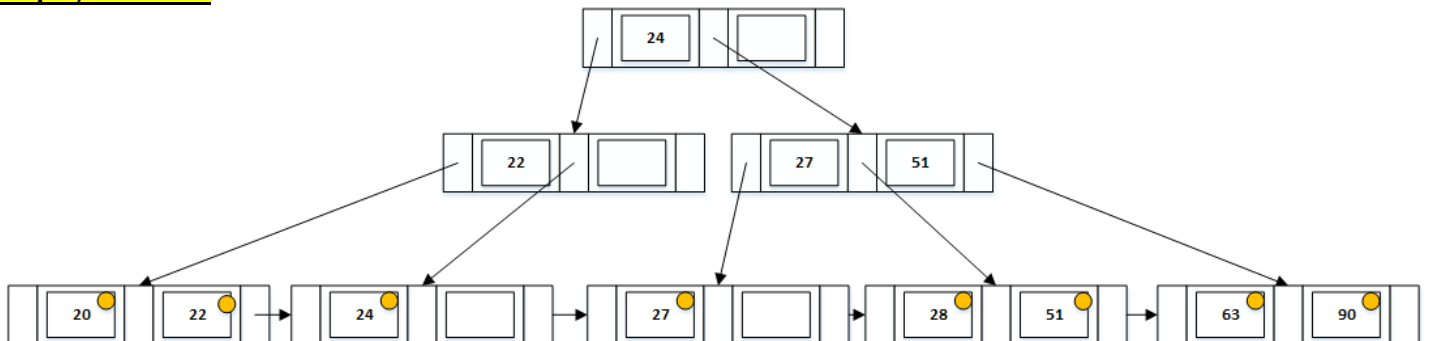
**Step 5) After inserting 20.**



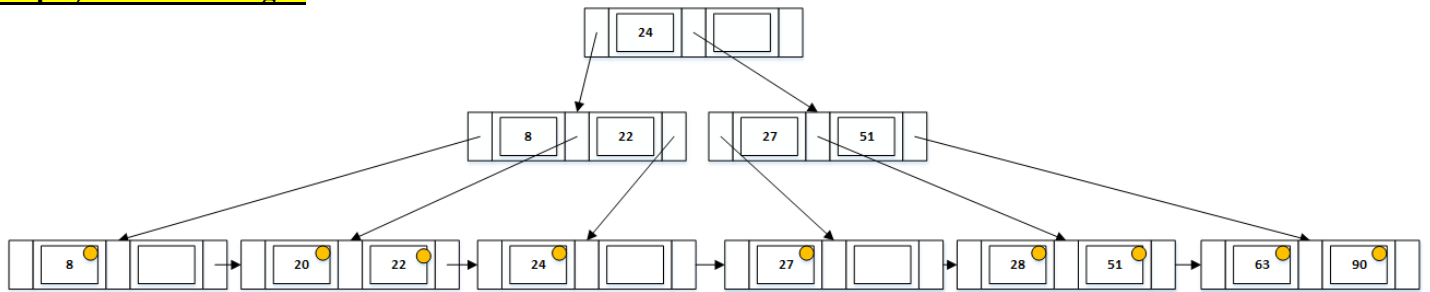
**Step 6) After inserting 51.**



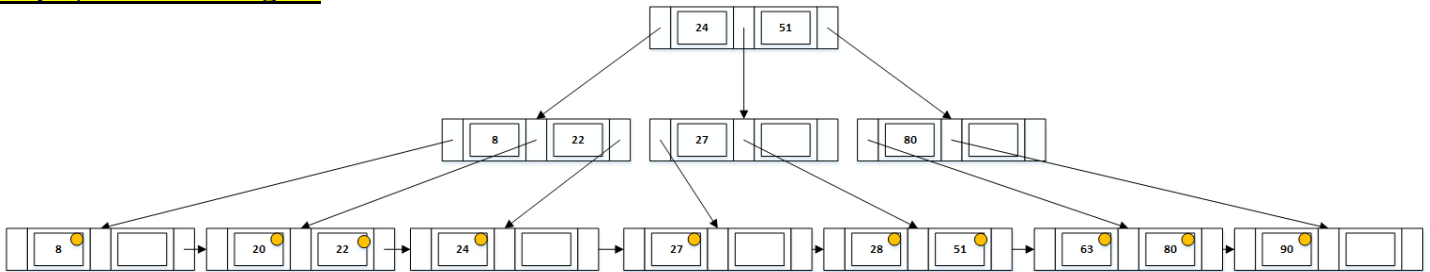
**Step 7) Insert 63.**



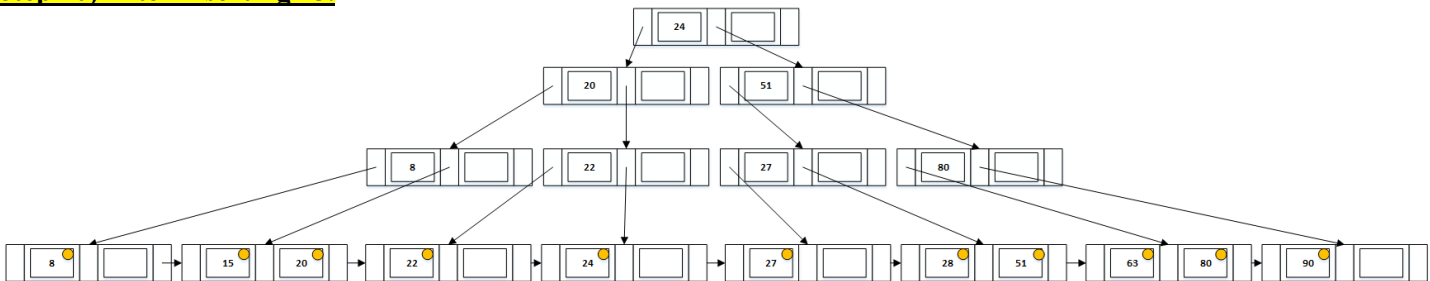
**Step 8) After inserting 8.**



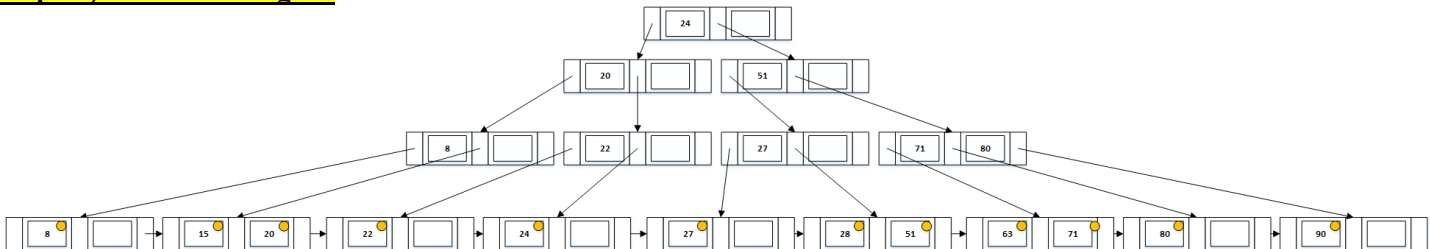
**Step 9) After inserting 80.**



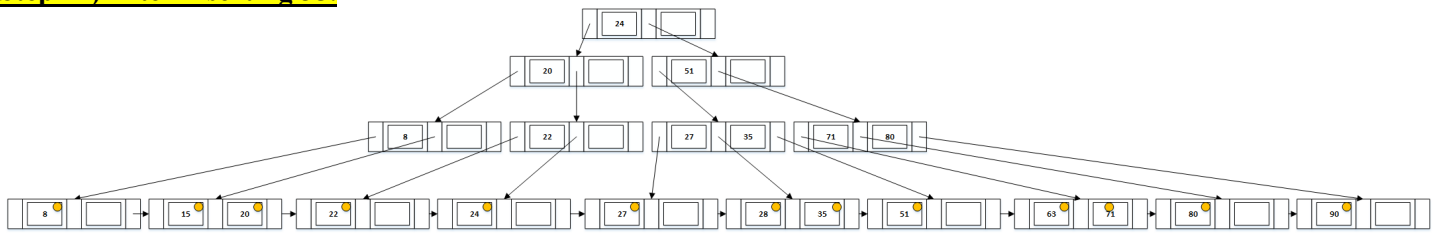
**Step 10) After inserting 15.**



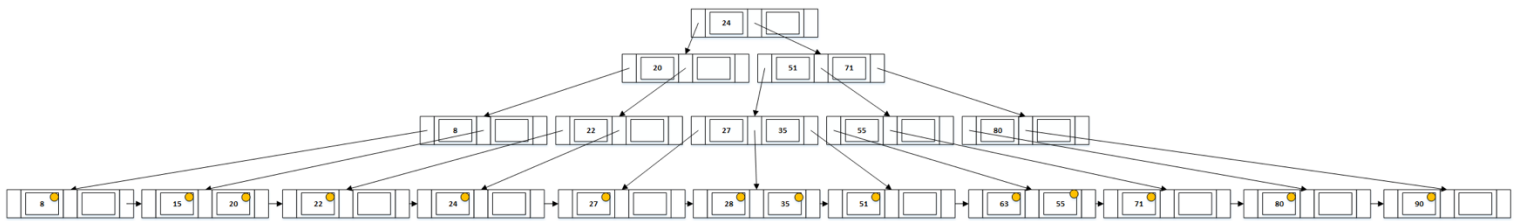
**Step 11) After inserting 71.**



**Step 12) After inserting 35.**

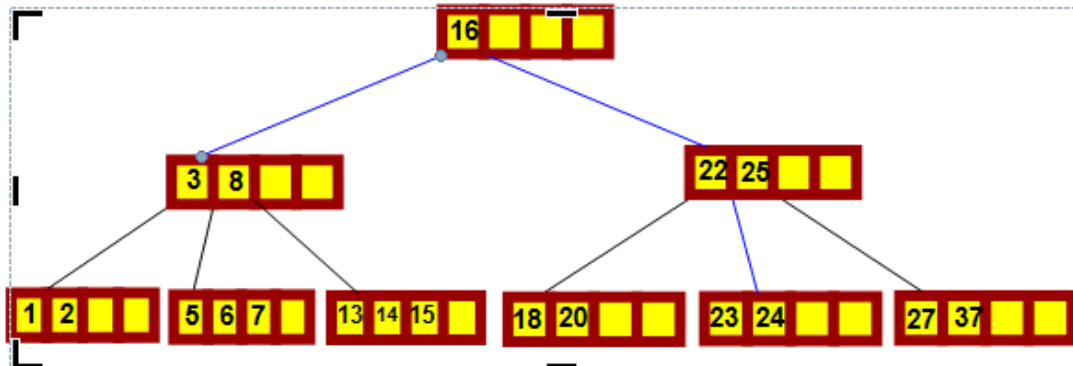


**Step 13) After inserting 55.**



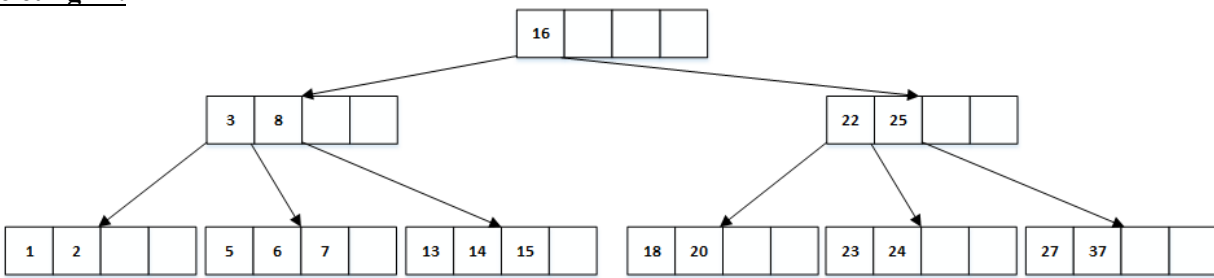
**Question 5:**

Consider the following B-tree.

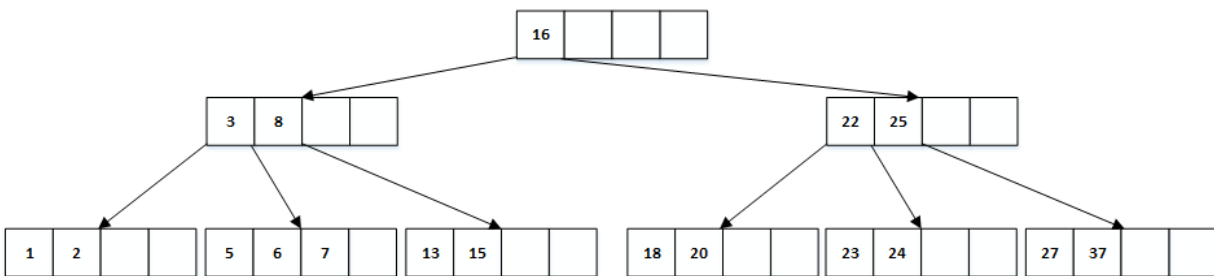


a) Redraw the tree after deleting 14.

Before deleting 14.

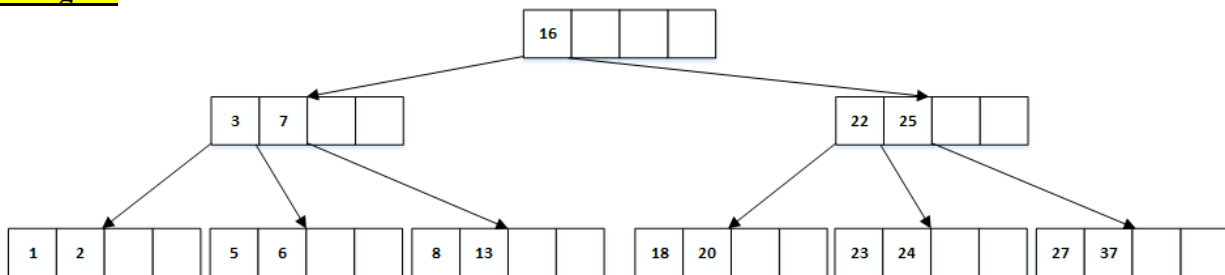


After deleting 14.



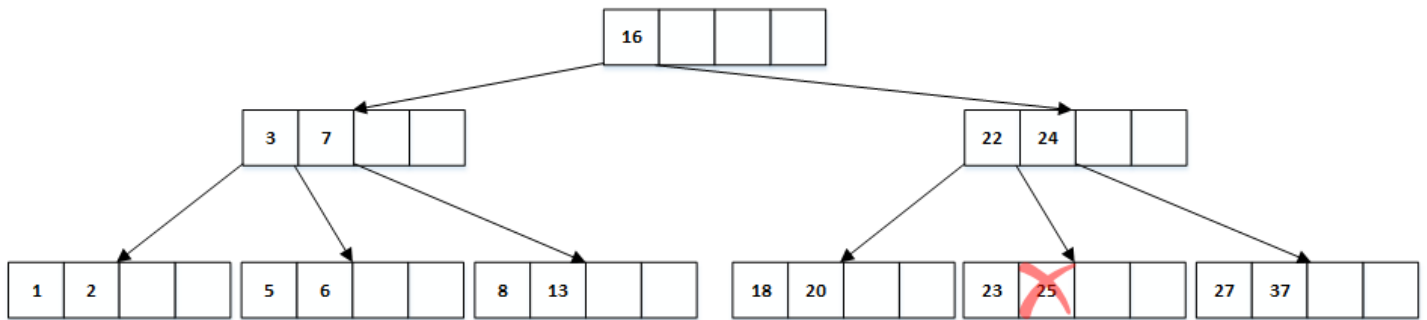
b) Again, redraw the tree after deleting 15.

After deleting 15.

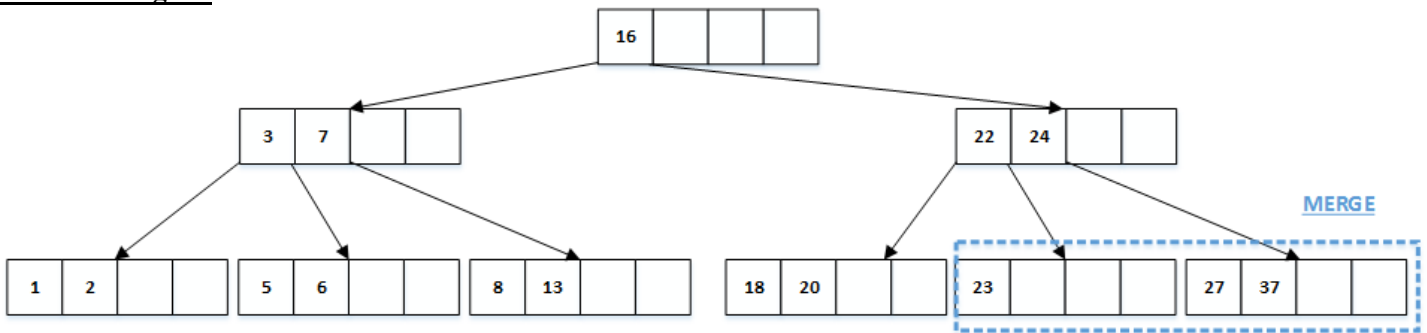


c) Again, redraw the tree after deleting 25.

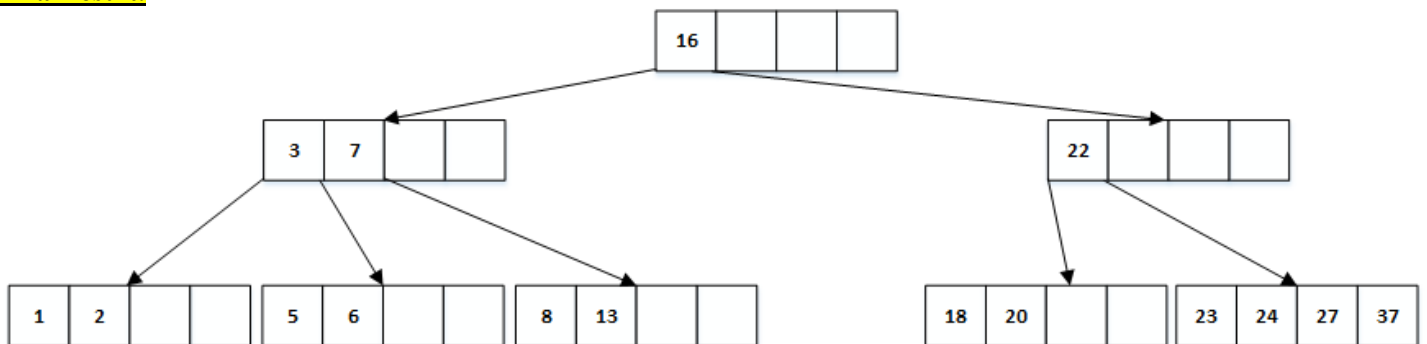
Swap 25 and 24. Delete 25.



After deleting 25.

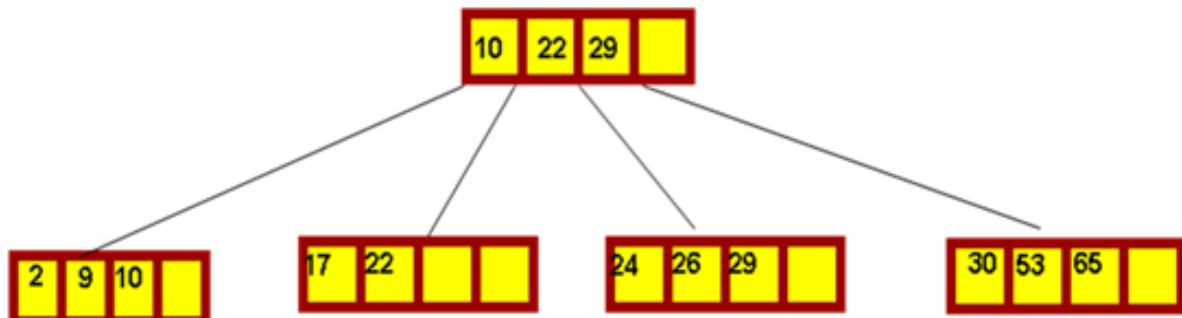


**Final result.**



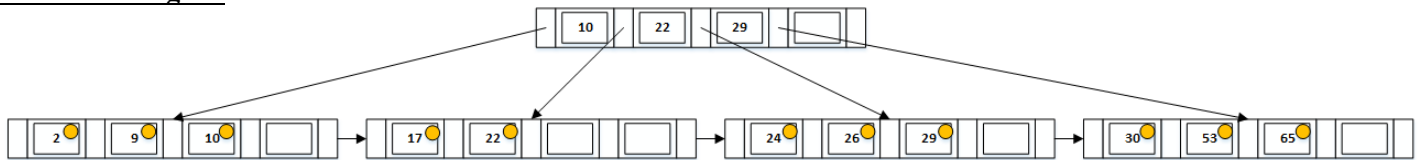
**Question 6:**

Consider the following B+ tree:

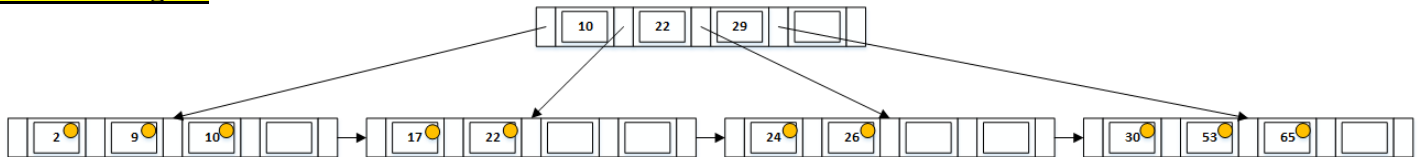


a) Delete 29 and redraw the tree after that

Before deleting 29.

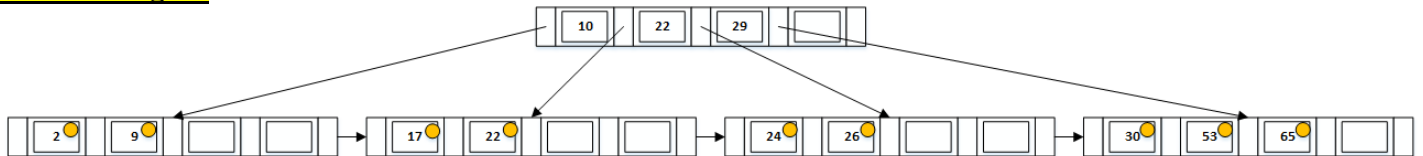


After deleting 29.



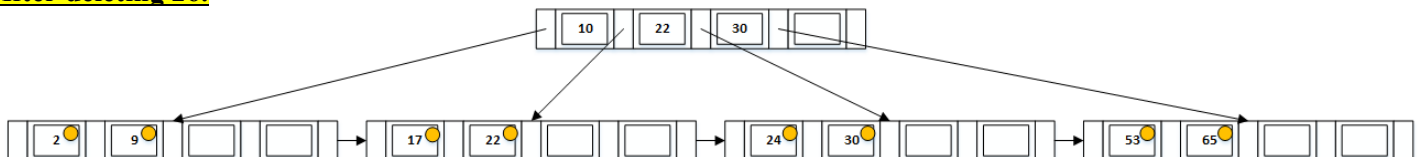
b) Next delete 10 and redraw the tree after that

After deleting 10.



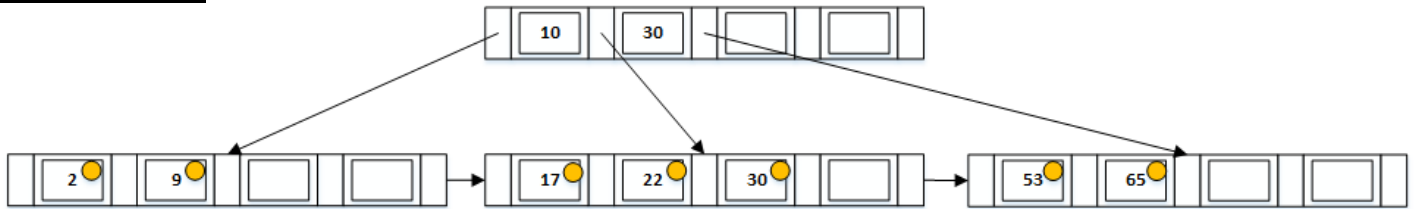
c) Next delete 26 and redraw the tree after that

After deleting 26.



d) Next delete 24 and redraw the tree after that

**After deleting 24.**



e) Next delete 9 and redraw the tree after that

**After deleting 9.**

