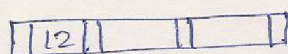


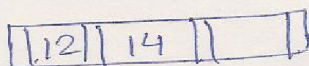
5. In B-tree the structure of Internal node and leaf node are ~~same~~ same. B-tree with order 4 means, maximum number of node pointer in a node is 4 and maximum number of search key in a node is 3.

The construction of B-tree for given search keys ~~are~~ is as follows.

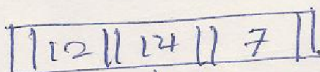
insert 12



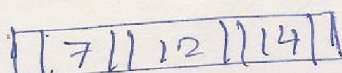
Insert 14



Insert 7



↓ in sorted order



Insert 32

The node can not contain more than 3 search key. So, the node will have to be splitted. The node can be splitted in two ways namely left biasing and right biasing. Depending on this there are two possibilities of tree to exist. Both are correct. Once a method is used for splitting we will have to use the same when ever we will split a node. Here we are using right biasing.

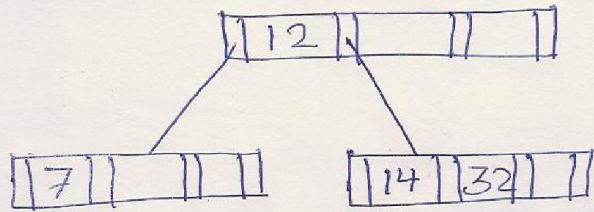
The search key in sorted order are

7, 12, 14, 32.

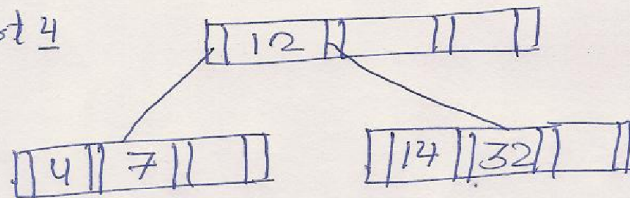
According to right biased splitting 12 will be the root with 7 ~~in~~ in



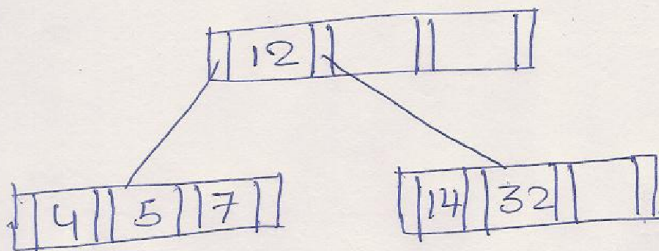
left node and 14 and 32 in right child node.



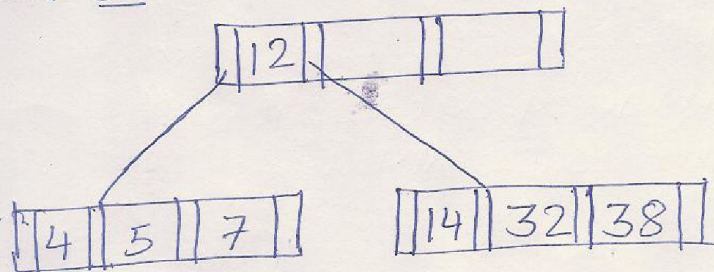
Insert 4



Insert 5



Insert 38



Insert 24

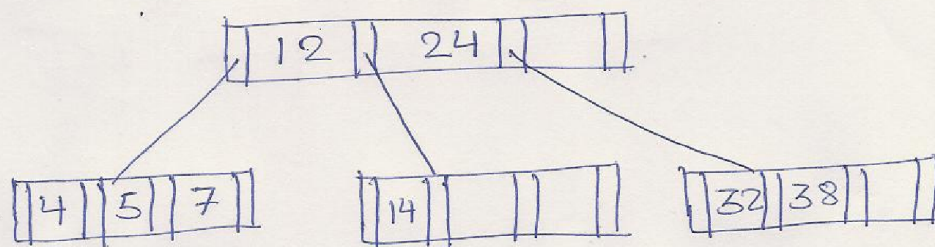
24 will be inserted into right child node where capacity of node is full and no siblings are having space so the node will have to be splitted. The splitting will be done using right busing method because we have used the same method for splitting the node once.



The search keys in sorted order are  
14, 24, 32, 38.

24 will become the root with 14 in left child node and 32 and 38 in right child node.

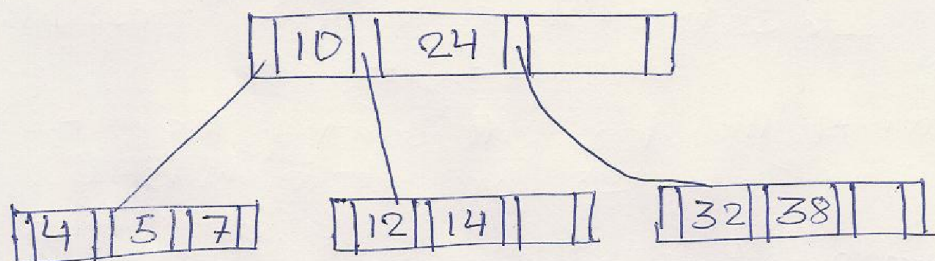
Here 24 is being sent to the above node. At root node search keys in sorted order are 7, 24.



Insert 10

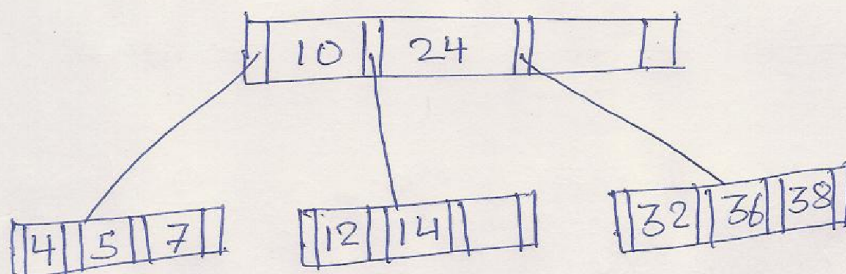
10 will be inserted in first left child node because  $10 < 12$ . The node is full. We should not go for node splitting till we have space available in sibling. Since in our case its sibling can accommodate two more search keys, we should go for key-redistribution.

The search keys in sorted order are 4, 5, 7, 10. So, 10 will be redistributed to its sibling (right). In this process '10' will be sifted to root at the place of 12 and 12 will be sifted to its right child and the search keys should be in sorted order.

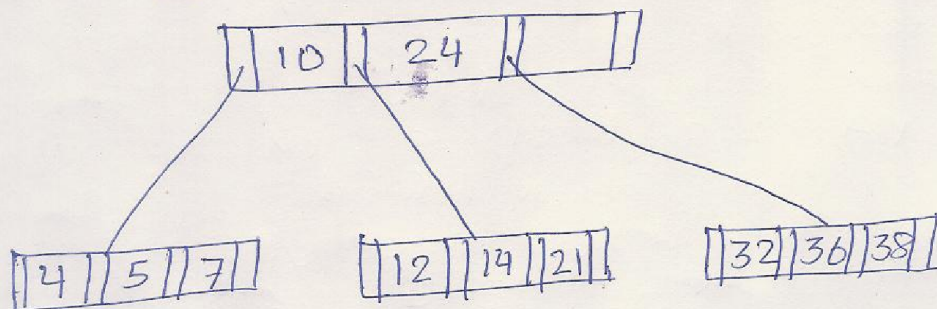


Insert 36

It will be inserted into right most child node (leaf) and the node will be sorted.



Insert 21



Insert 41

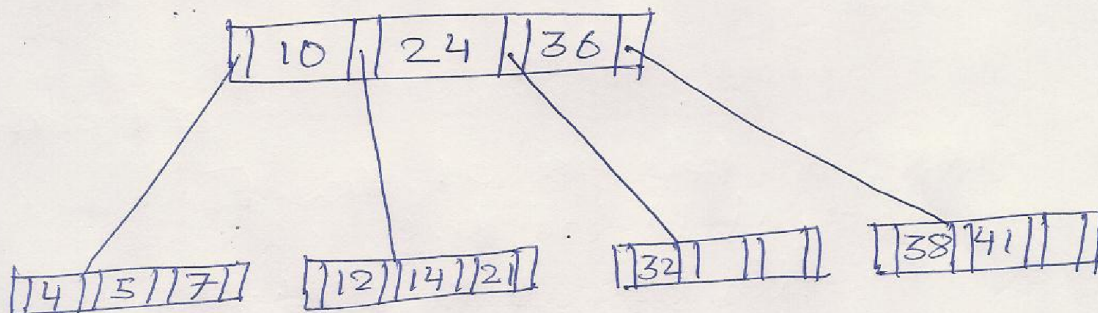
It will be inserted into right most child node. Since the node is full and also the siblings are full so, the node will have to be splitted.



The search keys in sorted order are

32, 36, 38, 41

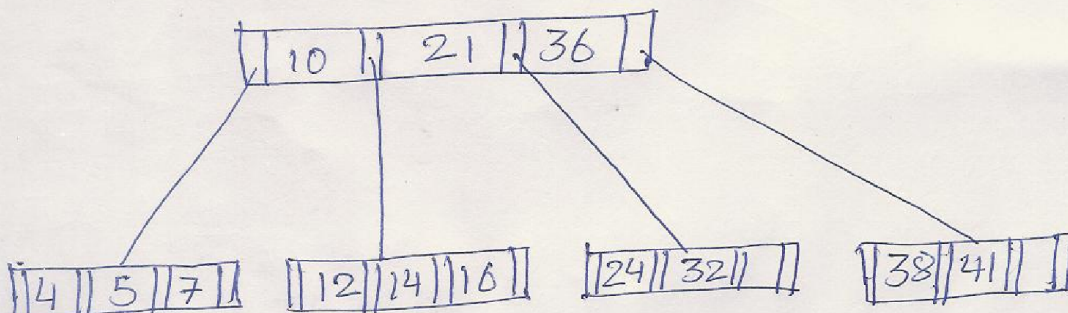
The key '36' will become the root.



Insert 16

It will be inserted in second leaf node because  $16 > 10$  and  $16 < 24$ . The node is full but space is available in next sibling. So, we will go for key redistribution.

The search keys in sorted order are 12, 14, 16, 21. '21' will be shifted.

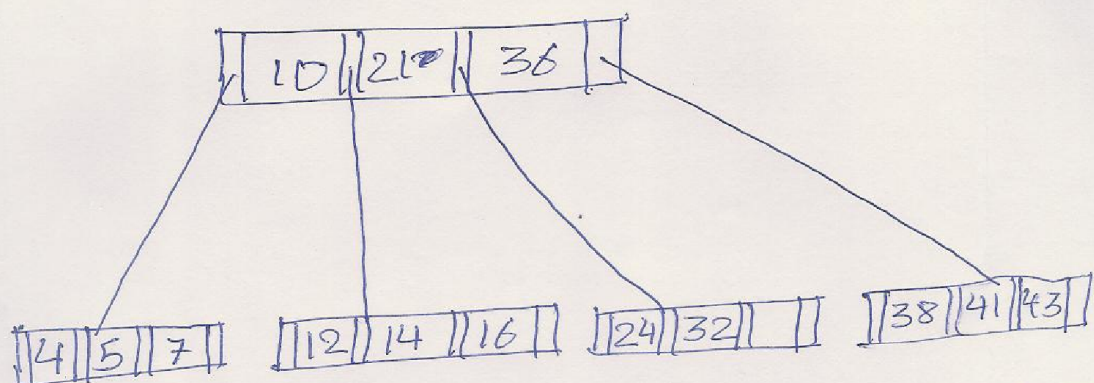




Insert 43

It will be inserted in last leaf node.

$$\underline{43 > 36}$$



This is the final tree. In the expected answer the students should have written the documentation of node splitting and key redistribution and should have drawn the tree after each step.

Ans-no-6-i)  $\pi_{sid, sname} (\sigma_{rating > 4} (suppliers))$

ii)  $\pi_{sid} (\sigma_{color='white'} (catalog \bowtie Parts))$

iii)  $\pi_{sname} (\sigma_{color='white'} (suppliers \bowtie catalog \bowtie Parts))$

iv)  $\pi_{sid} (\sigma_{color='red' OR color='white'} (catalog \bowtie Parts))$

v)  $\pi_{sid} (\sigma_{t1.sid = t2.sid AND t1.pid \neq t2.pid} (\rho(t1, catalog) \times \rho(t2, catalog)))$

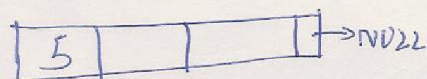


Ans-no-7 → In B<sup>+</sup>-tree the leaf node has record pointers but the internal nodes do not have record pointers. B<sup>+</sup>-tree with order 4 can have maximum 4 node pointers and maximum 3 search keys.

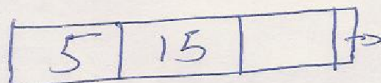
Multiple answers are possible depending on the method used in tree creation. Methods can be left biasing and right biasing. The search key which becomes the root when node is splitted will have to be present in either of the child node. Only one method should be followed throughout the tree creation.

A sample tree creation is as follows.

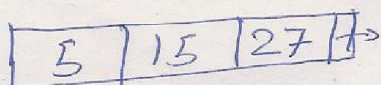
Insert 5



Insert 15



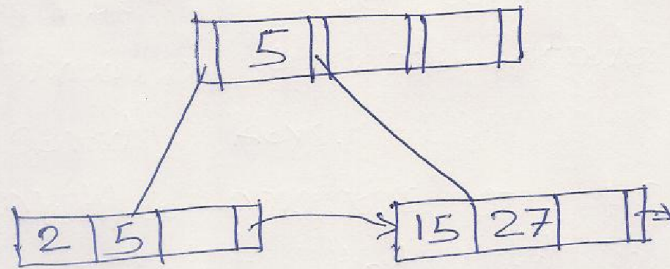
Insert 27





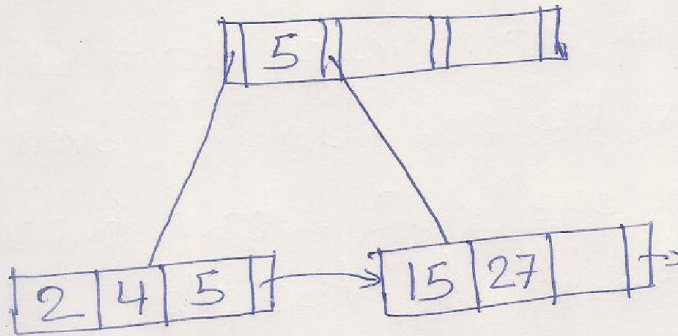
Insert 2

Right biased mode splitting.

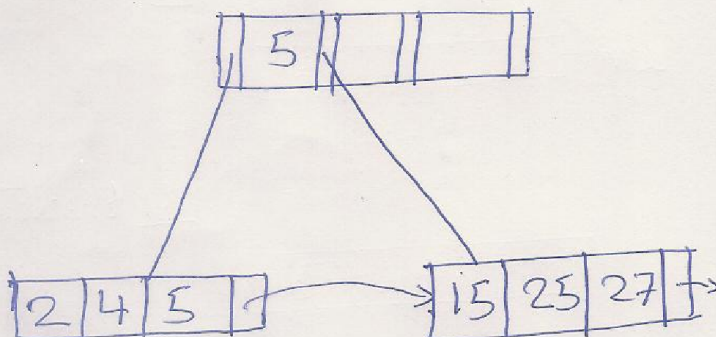


2, 5, 15, 27

Insert 4



Insert 25

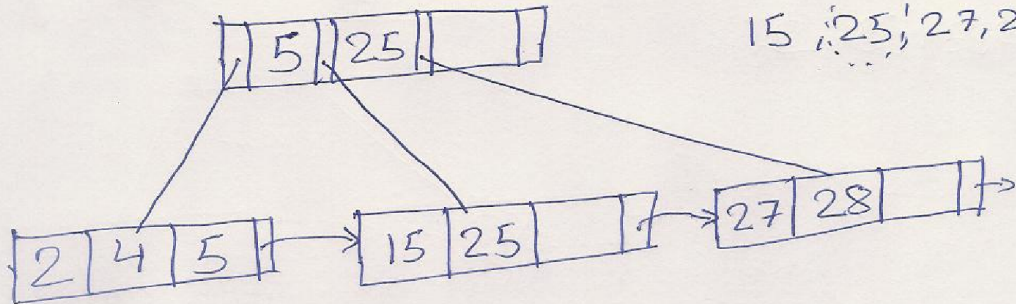




Insert 28

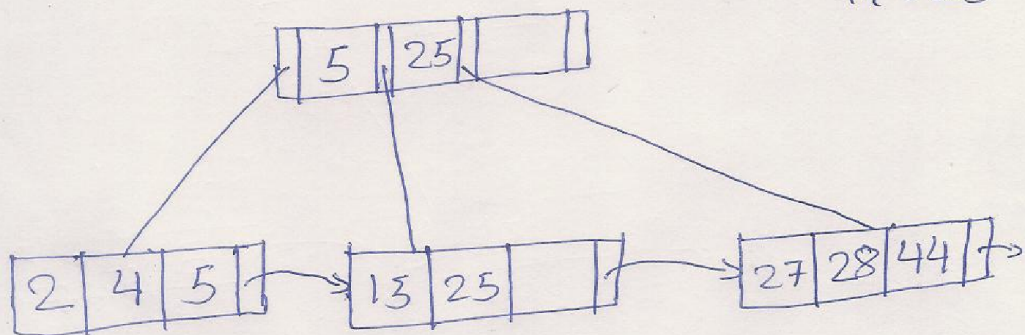
$28 > 5$

15, 25, 27, 28



Insert 44

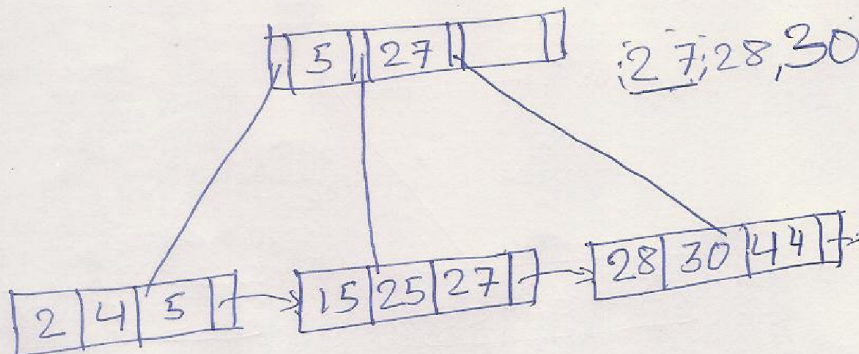
$44 > 25$



Insert 30

Key redistribution.

27, 28, 30, 44

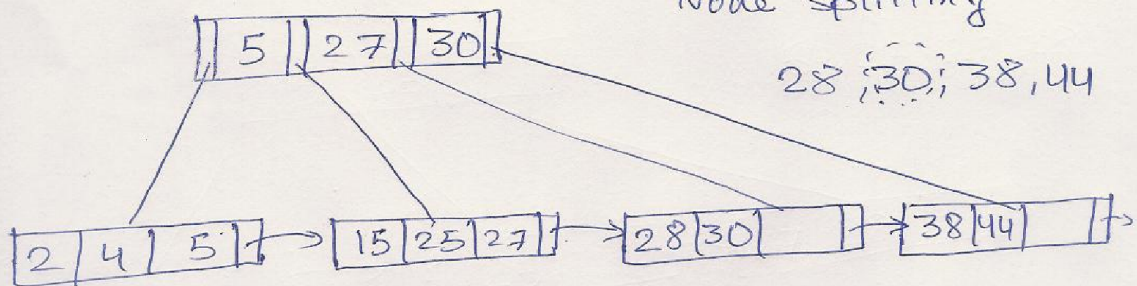


Insert 38

$$38 > 27$$

Node splitting

28, 30, 38, 44



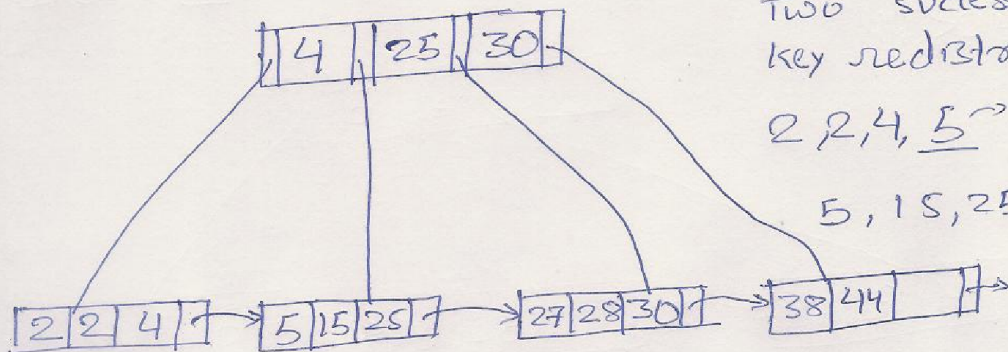
Insert 2

$$2 < 5$$

Two successive  
key redistribution.

2, 2, 4, 5

5, 15, 25, 27



Insert 0

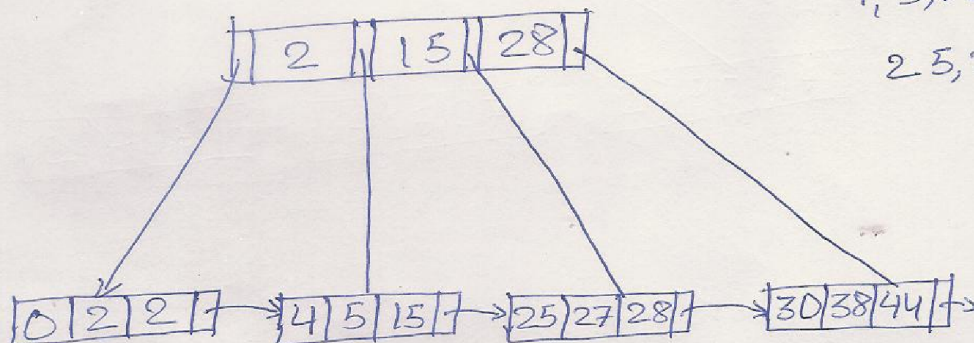
$$0 < 4$$

Three successive  
key redistribution

0, 2, 2, 4

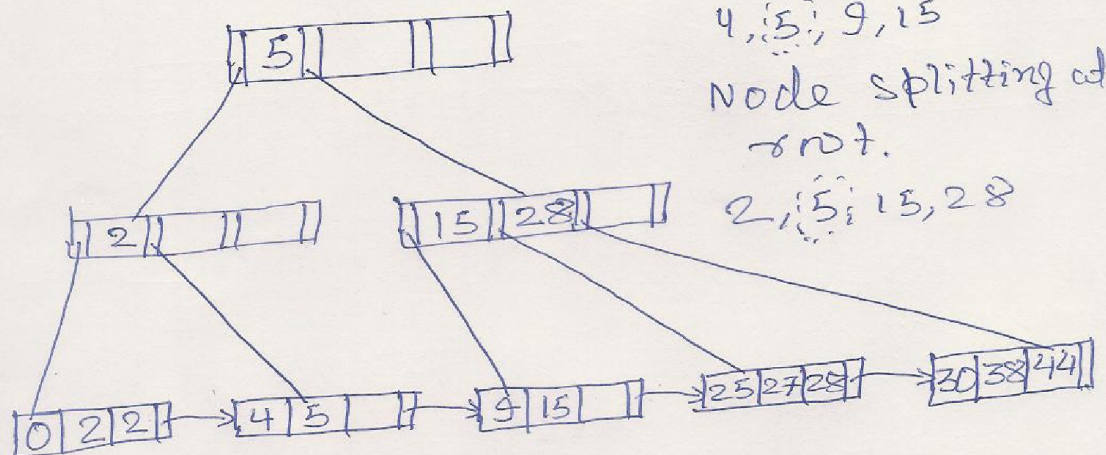
4, 5, 15, 25

25, 27, 28, 30





Insert 9



$2 < 9 < 15$   
Inserted in 2nd  
leaf node.

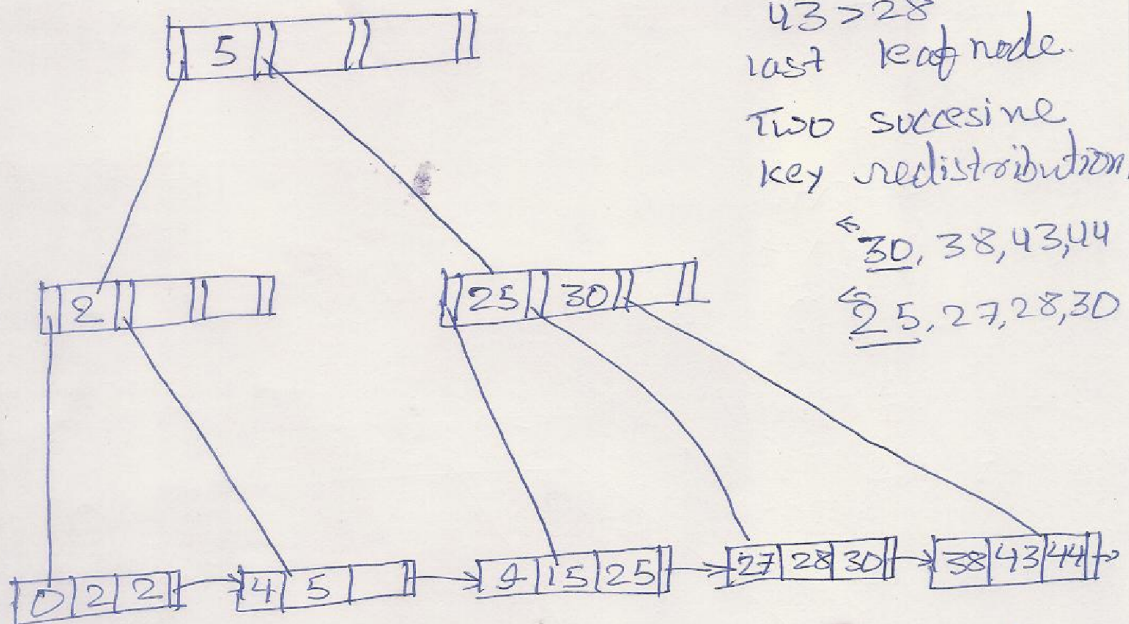
Node splitting.

4, 5, 9, 15

Node splitting at  
root.

2, 5, 15, 28

Insert 43



$43 > 5$

$43 > 28$

last leaf node.

Two successive  
key redistribution.

← 30, 38, 43, 44

← 5, 27, 28, 30

Using any method, if this tree is created,  
it will go to three levels. Above is the  
final tree created.

8. Following are the SQL query.

- i. 

```
SELECT PID
FROM PARTS
WHERE COLOR='green';
```
- ii. 

```
SELECT SID
FROM CATALOG NATURAL JOIN PARTS
WHERE COLOR= 'green';
```
- iii. 

```
SELECT SNAME,COLOR
FROM CATALOG NATURAL JOIN PARTS NATURAL JOIN SUPPLIERS
ORDER BY(COLOR);
```
- iv. 

```
SELECT SID
FROM CATALOG NATURAL JOIN PARTS
WHERE COLOR= 'red'
INTERSECT
SELECT SID
FROM CATALOG NATURAL JOIN PARTS
WHERE COLOR='green';
```
- v. 

```
SELECT C1.SID
FROM CATALOG C1, CATALOG C2
WHERE C1.SID=C2.SID AND C1.PID!=C2.PID;
```

9. Following are the answer of this question.

- i. Database is a collection of interrelated data files. These files will have set of similar records. Record is a set of attribute values. attribute is a basic property of an entity which is in interest of the table.
- ii. Database management system is a software which helps user to create and manipulate while keeping the database consistent and secure. Give some examples.
- iii. A number of advantages of applying database approach in application system are obtained including following: Students should have mentioned at least any four of these.
  - Control of data redundancy  
The database approach attempts to eliminate the redundancy by integrating the file. Although the database approach does not eliminate redundancy entirely, it controls the amount of redundancy inherent in the database.
  - Data consistency  
By eliminating or controlling redundancy, the database approach reduces the risk of inconsistencies occurring. It ensures all copies of the idea are kept consistent.
  - More information from the same amount of data  
With the integration of the operated data in the database approach, it may be possible to derive additional information for the same data.
  - Sharing of data  
Database belongs to the entire organization and can be shared by all authorized users.
  - Improved data integrity  
Database integrity provides the validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate.
  - Improved security



Database approach provides a protection of the data from the unauthorized users. It may take the term of user names and passwords to identify user type and their access right in the operation including retrieval, insertion, updating and deletion.

- **Enforcement of standards**  
The integration of the database enforces the necessary standards including data formats, naming conventions, documentation standards, update procedures and access rules.
- **Economy of scale**  
Cost savings can be obtained by combining all organization's operational data into one database with applications to work on one source of data.
- **Balance of conflicting requirements**  
By having a structural design in the database, the conflicts between users or departments can be resolved. Decisions will be based on the base use of resources for the organization as a whole rather than for an individual entity.
- **Improved data accessibility and responsiveness**  
By having an integration in the database approach, data accessing can be crossed departmental boundaries. This feature provides more functionality and better services to the users.
- **Increased productivity**  
The database approach provides all the low-level file-handling routines. The provision of these functions allows the programmer to concentrate more on the specific functionality required by the users. The fourth-generation environment provided by the database can simplify the database application development.
- **Improved maintenance**  
Database approach provides a data independence. As a change of data structure in the database will be affect the application program, it simplifies database application maintenance.
- **Increased concurrency**  
Database can manage concurrent data access effectively. It ensures no interference between users that would not result any loss of information nor loss of integrity.
- **Improved backing and recovery services**  
Modern database management system provides facilities to minimize the amount of processing that can be lost following a failure by using the transaction approach.
- **Disadvantages**  
In spite of a large number of advantages can be found in the database approach, it is not without any challenge. The following disadvantages can be found including:
  - **Complexity**  
Database management system is an extremely complex piece of software. All parties must be familiar with its functionality and take full advantage of it. Therefore, training for the administrators, designers and users is required.
  - **Size**  
The database management system consumes a substantial amount of main memory as well as a large number amount of disk space in order to make it run efficiently.
  - **Cost of DBMS**  
A multi-user database management system may be very expensive. Even after the installation, there is a high recurrent annual maintenance cost on the software.
  - **Cost of conversion**  
When moving from a file-base system to a database system, the company is required to have additional expenses on hardware acquisition and training cost.
  - **Performance**  
As the database approach is to cater for many applications rather than exclusively for a particular one, some applications may not run as fast as before.

- Higher impact of a failure

The database approach increases the vulnerability of the system due to the centralization. As all users and applications rely on the database availability, the failure of any component can bring operations to a halt and affect the services to the customer seriously.

- iv. Entity is any real world object which has a set of properties. Should have explained with help of example.

10. The given table is not normalized. The table has following attributes. sid, sname, cname, cfee. cname and cfee are multivalued attributes. For normalizing the table to 1NF we will have to decompose the table to two tables with any name say R1 and R2.

R1(sid, sname)    R2(sid, cname, cfee)

The table R1 is in 2NF but R2 is not. We further decompose the table R2 into R21 and R22 as follows.

R21(sid, cname)    R22(sid, cfee)

These tables are in 3NF. Students should draw these tables and show the reasons of not being in particular normal form. They will have to show whether the decomposition (if done) is lossless and dependency preserving.