Introduction to Information System

Section 4

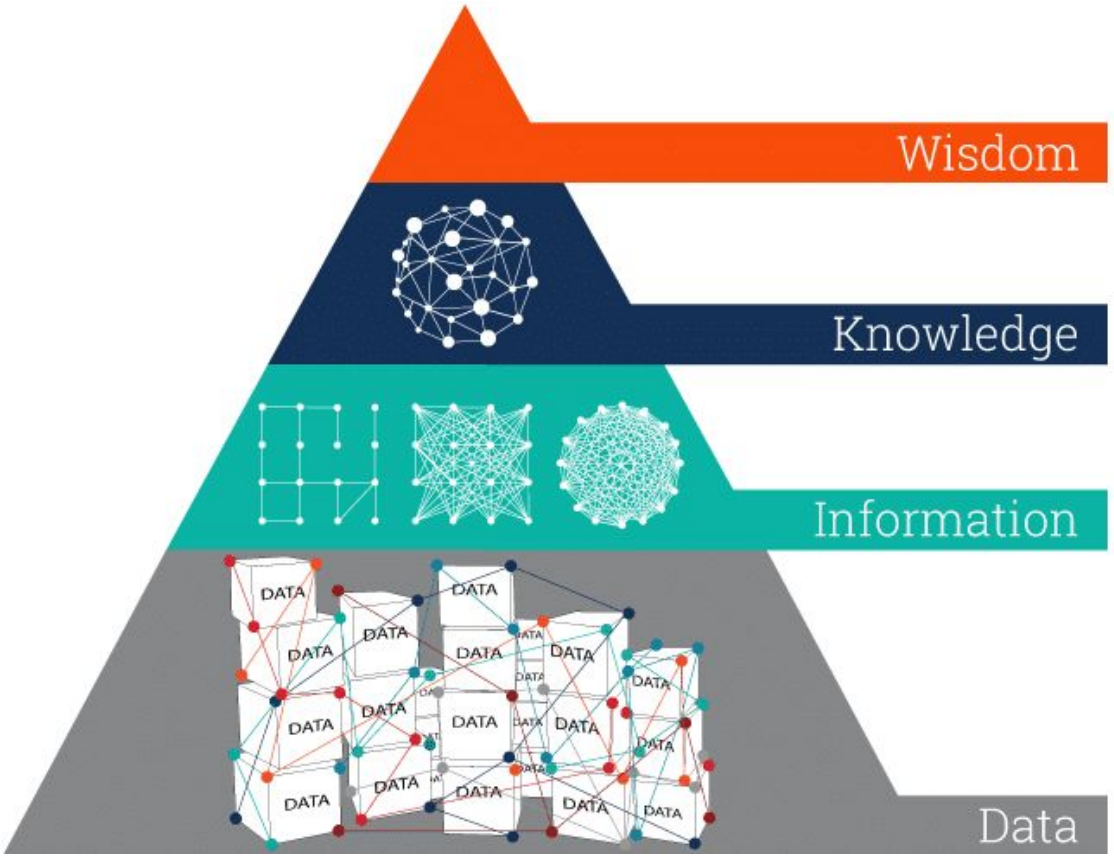
introduction Database
& DBMS

- **Introduction:**

  In computerized information system data are the basic resource of the organization. So, proper organization and management for data is required for organization to run smoothly. Database management system deals the knowledge of how data stored and managed on a computerized information system. In any organization, it requires accurate and reliable data for better decision making, ensuring privacy of data and controlling data efficiently.

  The examples include deposit and/or withdrawal from a bank, hotel, airline or railway reservation, purchase items from supermarkets in all cases, a database is accessed.

What is the Data, Information, Knowledge, Wisdom (DIKW) Pyramid?

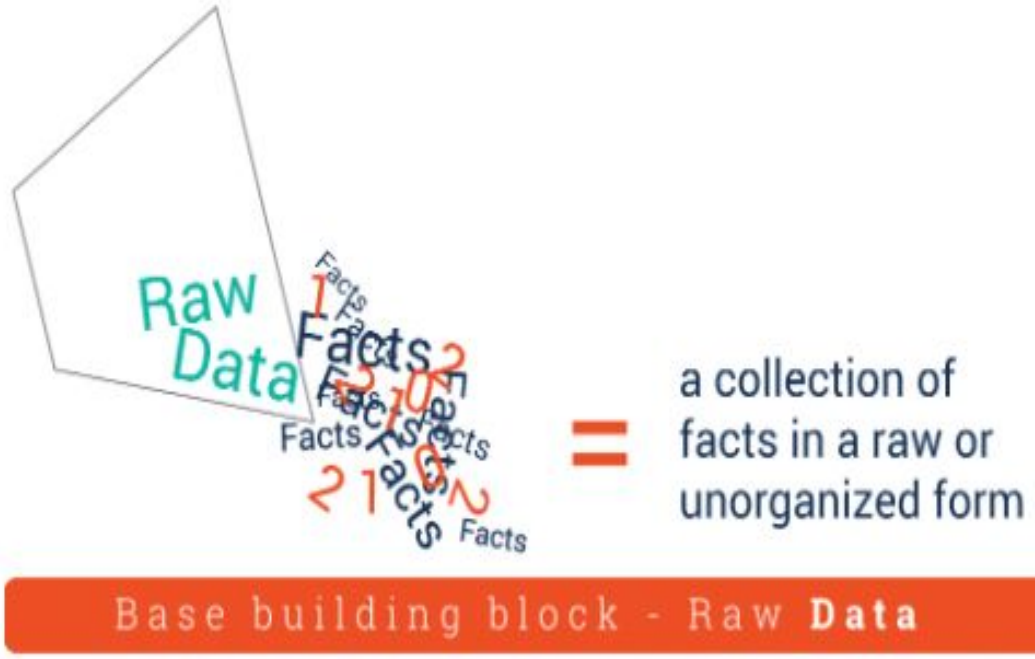


Each step up the pyramid answers questions about and adds value to the initial data.

How to Scale Data Up the Knowledge Pyramid

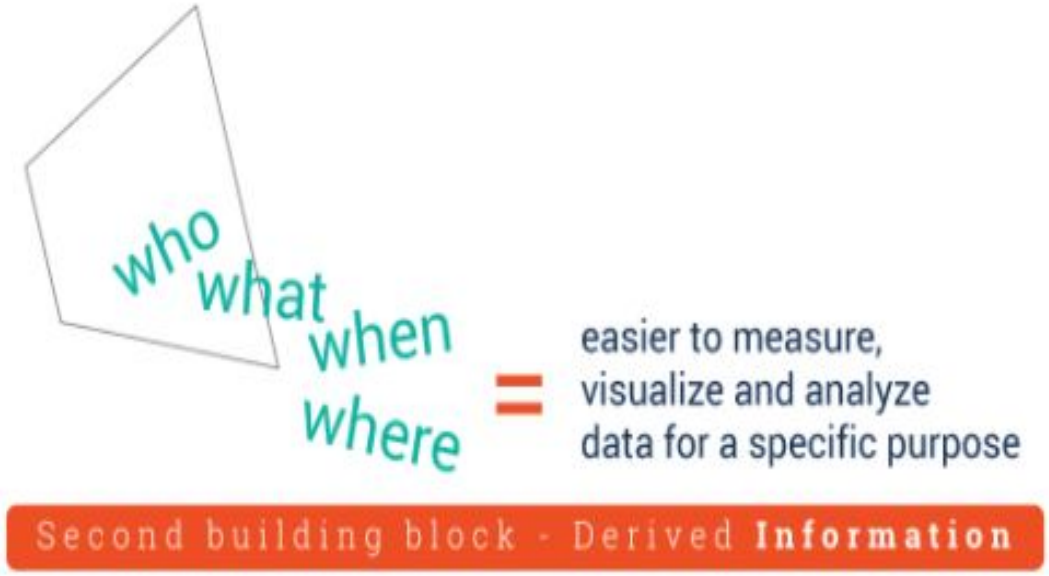
So, let's have a look at the individual components of the Knowledge Pyramid and how we move from one to the next.

Data



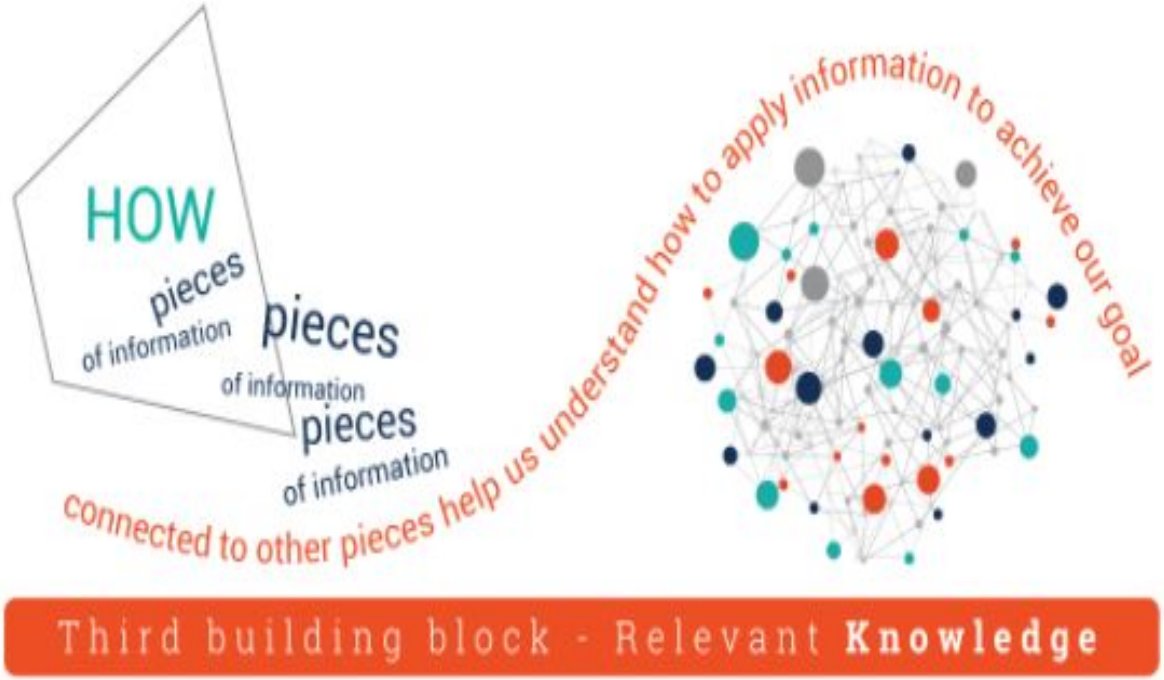
Data is a collection of facts in a raw or unorganized form such as numbers or characters.

Information



Information is the next building block of the DIKW Pyramid. This is data that has been "cleaned" of errors and further processed in a way that makes it easier to measure, visualize and analyze for a specific purpose.

Knowledge



Third building block - Relevant Knowledge

"How" is the information, derived from the collected data, relevant to our goals? "How" are the pieces of this information connected to other pieces to add more meaning and value? And, maybe most importantly, "how" can we apply the information to achieve our goal?

Wisdom

Wisdom is the top of the DIKW hierarchy and to get there, we must answer questions such as 'why do something' and 'what is best'. In other words, wisdom is knowledge applied in action.

- **What is data?**

  Data are the known facts or figures that have implicit meaning. It can also be defined as it is the representation of facts, concepts or instructions in a formal manner, which is suitable for understanding and processing.

- Data can be represented in alphabets (A-Z, a-z), digits (0-9) and using special characters (+,-.#,$, etc) e.g: 25, “ajit” etc.

- **Information:**

  Information is the **processed data** on which decisions and actions are based. Information can be defined as the organized and classified data to provide meaningful values.

  Eg: “The age of Ravi is 25”

- **File:**

  File is a collection of related data stored in secondary memory.

- **Database:**

  A database is organized collection of related data of an organization stored in formatted way which is shared by multiple users.

  The main feature of data in a database are:

  1. It must be well organized
  2. It is related
  3. It is accessible in a logical order without any difficulty
  4. It is stored only once

- **Why Database:**

In order to overcome the limitation of a file system, a new approach was required. Hence a database approach emerged. A database is a persistent collection of logically related data. The initial attempts were to provide a centralized collection of data. A database has a self describing nature. It contains not only the data sharing and integration of data of an organization in a single database.

A small database can be handled manually but for a large database and having multiple users it is difficult to maintain it. In that case a computerized database is useful.

The advantages of database system over traditional, paper based methods of record keeping are:

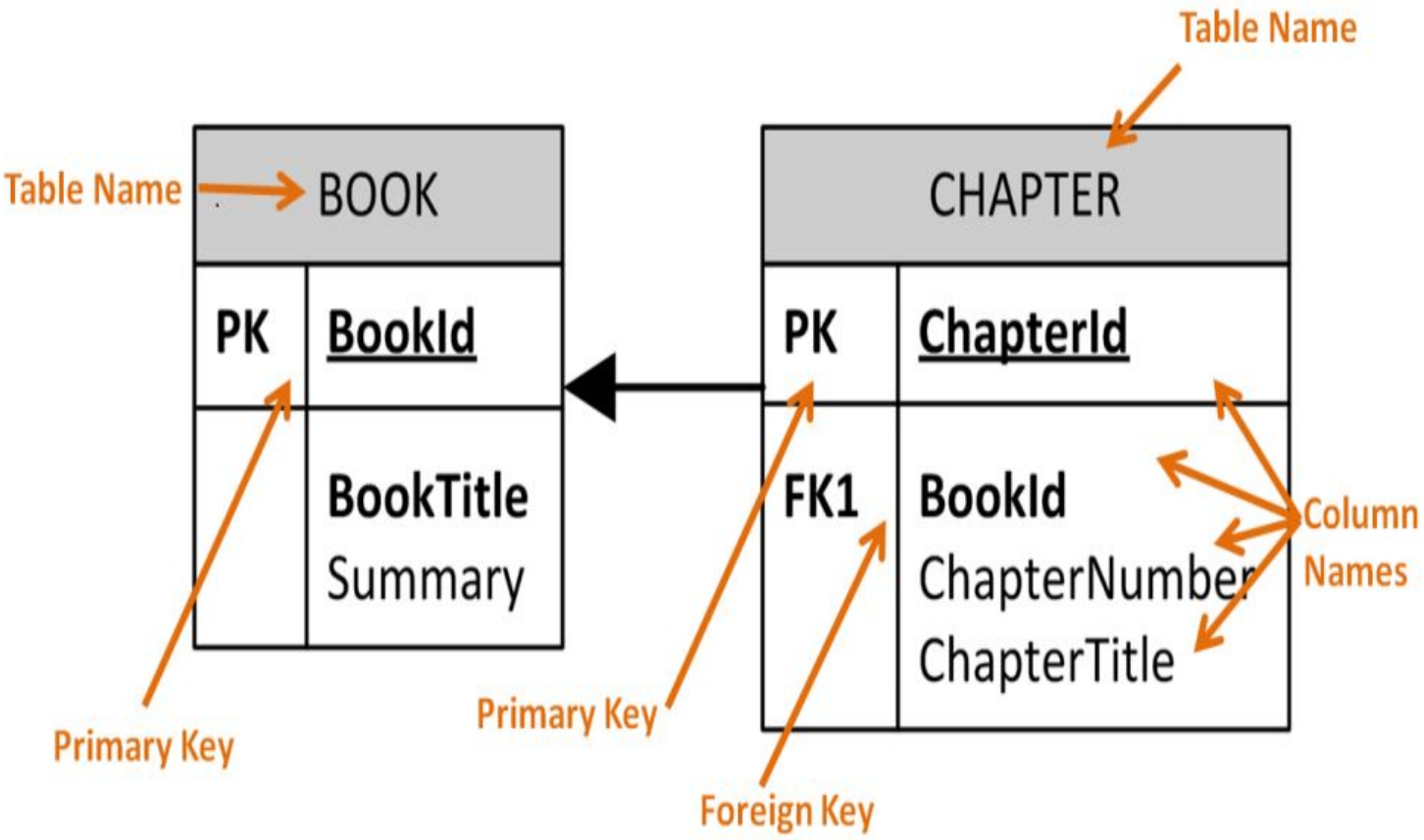
Compactness: No need for large amount of paper files

Speed: The machine can retrieve and modify the data more faster way then human being.

Less drudgery: Much of the maintenance of files by hand is eliminated.

Accuracy: Accurate, up-to-date information is fetched as per requirement of the user at any time.

- Tables : The basic units in a database are tables and the relationship between them. Strictly, a relational database is a collection of relations (frequently called tables). Below we see how a relationship between two tables are defined using Primary Keys and Foreign Keys.



Database as one table

Student_ID	Name	Contact	College	Course	Rank
100	Himanshu	7300934851	GEU	Btech	1
101	Ankit	7900734858	GEU	Btech	1
102	Aysuh	7300936759	GEU	Btech	1
103	Ravi	7300901556	GEU	Btech	1

The main problem in this table is :

Redundancy.

It means having multiple copies of same data in the database. This problem arises when a database is not normalized. Suppose a table of student details attributes are: student Id, student name, contact, college name, college rank.

As it can be observed that values of attribute college name, college rank, course is being repeated which can lead to problems. Problems caused due to redundancy are: Insertion anomaly, Deletion anomaly, and Updating anomaly.

- **Insertion Anomaly –**
  If a student detail has to be inserted whose course is not being decided yet then insertion will not be possible till the time course is decided for student. This problem happens when the insertion of a data record is not possible without adding some additional unrelated data to the record.
- **Deletion Anomaly –**

If the details of students in this table are deleted then the details of college will also get deleted which should not occur by common sense. This anomaly happens when deletion of a data record results in losing some unrelated information that was stored as part of the record that was deleted from a table. It is not possible to delete some information without loosing some other information in the table as well.

- **Updating Anomaly –**

Suppose if the rank of the college changes then changes will have to be all over the database which will be time-consuming and computationally costly. If updating does not occur at all places then database will be in inconsistent state.

Student_ID	Name	Contact	College	Course	Rank
100	Himanshu	7300934851	GEU	Btech	1
101	Ankit	7900734858	GEU	Btech	1
102	Aysuh	7300936759	GEU	Btech	1
103	Ravi	7300901556	GEU	Btech	1

- The solution is breaking the large table down small tables and make relations among tables.
- This action will reduce the Redundancy.

- **Database Management System (DBMS):**

  A database management system consists of collection of related data and refers to a set of programs for defining, creation, maintenance and manipulation of a database.

Functions of DBMS:

1. Defining database schema: it must give facility for defining the database structure also specifies access rights to authorized users.

2. Manipulation of the database: The dbms must have functions like insertion of record into database, updating of data, deletion of data, retrieval of data

3. Sharing of database: The DBMS must share data items for multiple users by maintaining consistency of data.

4. Protection of database: It must protect the database against unauthorized users.

5. Database recovery: If for any reason the system fails DBMS must facilitate data base recovery.

- **Advantages of DBMS:**

  Reduction of redundancies:

  Centralized control of data by the DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required avoiding duplication in the elimination of the inconsistencies that tend to be present in redundant data files.

  Sharing of Data:

  A database allows the sharing of data under its control by any number of application programs or users.

  Data Integrity:

  Data integrity means that the data contained in the database is both accurate and consistent. Therefore data values being entered for storage could be checked to ensure that they fall with in a specified range and are of the correct format.

  Data Security:

  The DBA who has the ultimate responsibility for the data in the dbms can ensure that proper access procedures are followed including proper authentication to access to the DataBase System and additional check before permitting access to sensitive data.

- **Conflict Resolution:**

  DBA resolve the conflict on requirements of various user and applications. The DBA chooses the best file structure and access method to get optional performance for the application.

  Data Independence:

  Data independence is usually considered from two points of views; physically data independence and logical data independence.

  Physical Data Independence allows changes in the physical storage devices or organization of the files to be made without requiring changes in the conceptual view or any of the external views and hence in the application programs using the data base.

  Logical Data Independence indicates that the conceptual schema can be changed without affecting the existing external schema or any application program.

- ## Disadvantage of DBMS:

  1. DBMS software and hardware (networking installation) **cost** is high.

  2. The **processing overhead** by the dbms for implementation of security, integrity and sharing of the data.

  3. **Centralized** database control.

  4. Setup of the database system **requires** more knowledge, money, skills, and time.

  5. Database management system (DBMS) is so **complex** for non-technical users. So, it isn't easy to manage and maintain database systems. Therefore, training for the designers, users, and administrators is necessary to efficiently run the database systems.

1. Database Management Systems

There are Database Management Systems (DBMS), such as:

- Microsoft SQL Server
- Oracle
- Sybase
- dBase
- Microsoft Access
- MySQL from Sun Microsystems (Oracle)
- DB2 from IBM
- etc.

- **What are indexes?**

  Indexing mechanisms used to speed up access to desired data.

  E.g., author catalog in library

- An index is a database structure that you can use to improve the performance of database activity. A database table can have one or more indexes associated with it.

- Indexes are a powerful tool used in the background of a database to speed up querying. Indexes power queries by providing a method to quickly lookup the requested data.
- Simply put, an index is a pointer to data in a table. An index in a database is very similar to an index in the back of a book.

- **How are indexes created?**

In a database, data is stored in rows which are organized into tables. Each row has a unique key which distinguishes it from all other rows and those keys are stored in an index for quick retrieval.

- Since keys are stored in indexes, each time a new row with a unique key is added, the index is automatically updated. However, sometimes we need to be able to quickly lookup data that is not stored as a key. For example, we may need to quickly lookup **customers** by telephone number. It would not be a good idea to use a unique constraint because we can have multiple customers with the same phone number. In these cases, we can create our own indexes.
- The syntax for creating an index will vary depending on the database. However, the syntax typically includes a CREATE keyword followed by the INDEX keyword and the name we'd like to use for the index. Next should come the ON keyword followed by the name of the table that has the data we'd like to quickly access. Finally, the last part of the statement should be the name(s) of the columns to be indexed.
- CREATE INDEX customers_by_phone ON customers (phone_number)

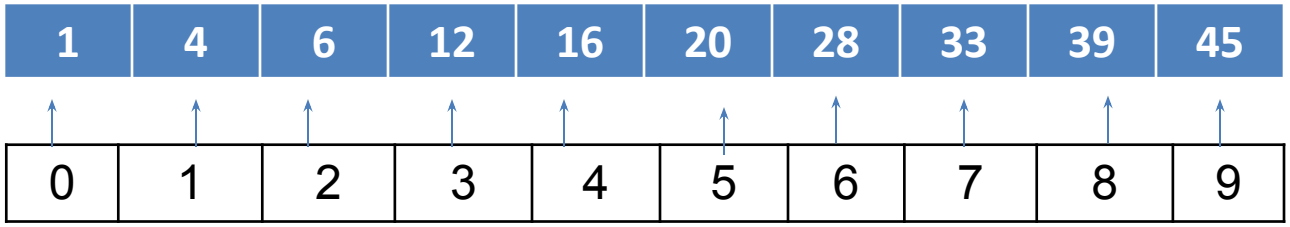
This example, if we would like to index phone numbers from a customers table.

- The users cannot see the indexes, they are just used to speed up searches/queries.
- Updating a table with indexes takes more time than updating a table without (because the indexes also need an update). So, only create indexes on columns that will be frequently searched against.

- **What is a linear search?**

A linear search is also known as a sequential search that simply scans each element at a time. Suppose we want to search an element in an array or list; we simply calculate its length and do not jump at any item.

- **Let's consider a simple example.**
- **Suppose we have an array of 10 elements as shown in the below figure:**



The above figure shows an array of Numeric type having 10 values. If we want to search (33), then the searching begins from the 0^{th} element and scans each element until the element, i.e., (33) is found. We cannot directly jump from the 0^{th} element to the 7^{th} element, i.e., each element is scanned one by one till the element is found or the end.

Complexity of Linear search

As linear search scans each element one by one until the element is found. If the number of elements increases, the number of elements to be scanned is also increased. We can say that the ***time taken to search the elements is proportional to the number of elements***. Therefore, the worst-case complexity is **O(n)**

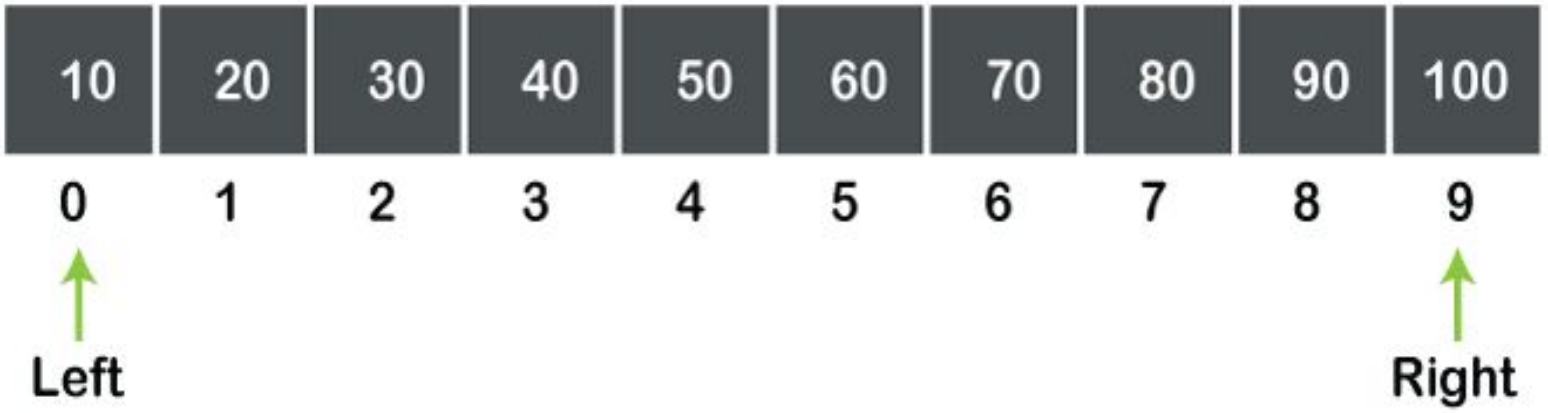
- **What is a Binary search?**
- A binary search is a search in which the middle element is calculated to check whether it is smaller or larger than the element which is to be searched. The main advantage of using binary search is that it does not scan each element in the list. Instead of scanning each element, it performs the searching to the half of the list. So, the binary search takes less time to search an element as compared to a linear search.
- The one pre-requisite of binary search is that an array should **be in sorted order**, whereas the linear search works on both sorted and unsorted array. The binary search algorithm is based on the divide and conquer technique, which means that it will divide the array recursively.
- There are three cases used in the binary search:
- **Case 1**: data > a[mid] then left = mid+1.
- **Case 2**: data < a[mid] then right=mid-1.
- **Case 3**: data = a[mid] // element is found.

In the above case, '**a**' is the name of the array, **mid** is the index of the element calculated recursively, **data** is the element that is to be searched, **left** denotes the left element of the array and **right** denotes the element that occur on the right side of the array.

- **Let's understand the working of binary search through an example.**
- Suppose we have an array of 10 size which is indexed from 0 to 9 as shown in the below figure:

We want to search for 70 element from the above array.

Step 1: First, we calculate the middle element of an array. We consider two variables, i.e., left and right. Initially, left =0 and right=9 as shown in the below figure:



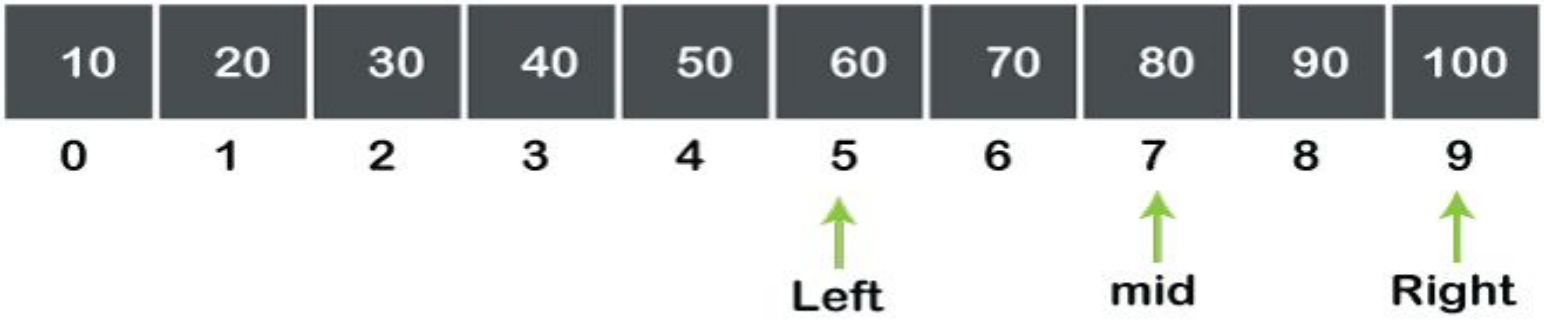
The middle element value can be calculated as:

$$mid = \frac{left + right}{2}$$

Therefore, mid = 4 and a[mid] = 50. The element to be searched is 70, so a[mid] is not equal to data. The case 2 is satisfied, i.e., data>a[mid].

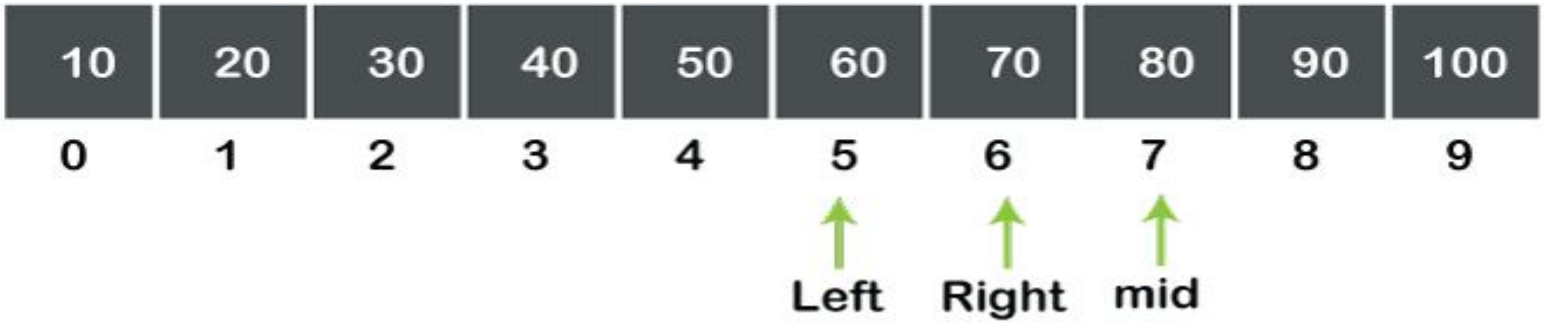
Step 2: As data>a[mid], so the value of left is incremented by mid+1, i.e., left=mid+1. The value of mid is 4, so the value of left becomes 5. Now, we have got a subarray as shown in the below figure:

Now again, the mid-value is calculated by using the above formula, and the value of mid becomes 7. Now, the mid can be represented as:



In the above figure, we can observe that a[mid]>data, so again, the value of mid will be calculated in the next step.

Step 3: As a[mid]>data, the value of right is decremented by mid-1. The value of mid is 7, so the value of right becomes 6. The array can be represented as:



The value of mid will be calculated again. The values of left and right are 5 and 6, respectively. Therefore, the value of mid is 5. Now the mid can be represented in an array as shown below:

In the above figure, we can observe that a[mid]<data.

Step 4: As a[mid]<data, the left value is incremented by mid+1. The value of mid is 5, so the value of left becomes 6.

Now the value of mid is calculated again by using the formula which we have already discussed. The values of left and right are 6 and 6 respectively, so the value of mid becomes 6 as shown in the below figure:

We can observe in the above figure that a[mid]=data. Therefore, the search is completed, and the element is found successfully.

of comparison	Linear search	Binary search
Definition	The linear search starts searching from the first element and compares each element with a searched element till the element is not found.	It finds the position of the searched elem the middle element of the ar
Sorted data	In a linear search, the elements don't need to be arranged in sorted order.	The pre-condition for the binary searc elements must be arranged in a so
nplementation	The linear search can be implemented on any linear data structure such as an array, linked list, etc.	The implementation of binary search is l be implemented only on those data struc two-way traversal.
Approach	It is based on the sequential approach.	It is based on the divide and conque
Size	It is preferable for the small-sized data sets.	It is preferable for the large-size d
Efficiency	It is less efficient in the case of large-size data sets.	It is more efficient in the case of large-s
st-case scenario	In a linear search, the worst- case scenario for finding the element is O(n).	In a binary search, the worst-case scen the element is $O(\log_2 n)$.
t-case scenario	In a linear search, the best-case scenario for finding the first element in the list is O(1).	In a binary search, the best-case scena the first element in the list is
mensional array	It can be implemented on both a	It can be implemented only on a multidim

Be ready for any quiz about this slide at the next week.

Thanks For You