DATA STRUCTURES AND ALGORITHMS.

Beginner to Advance.

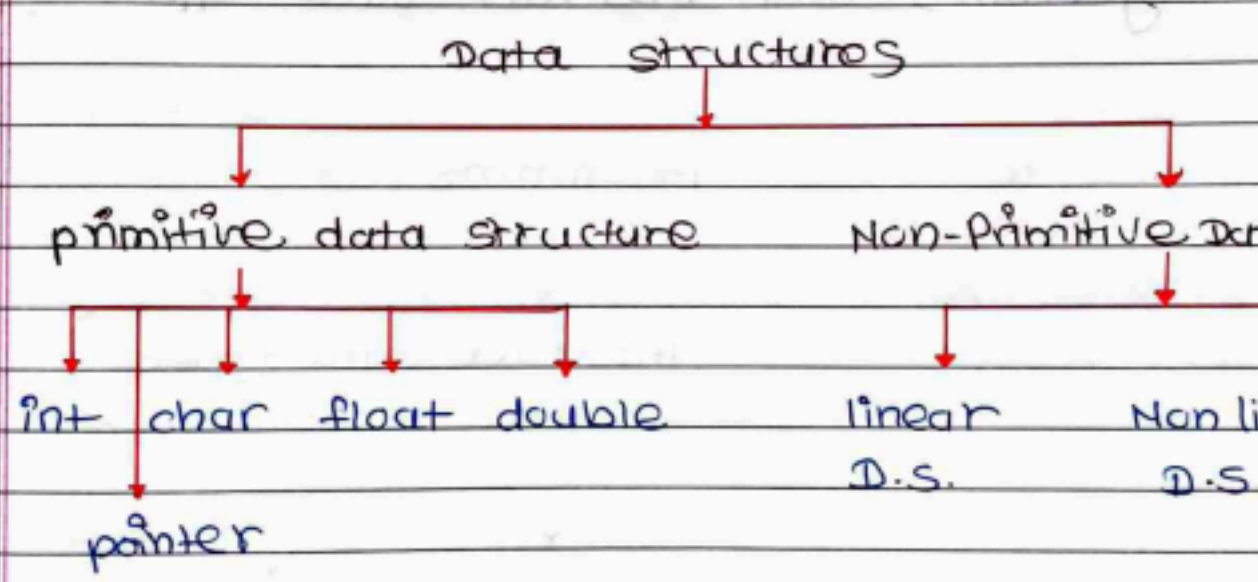
INDEX

What is Data structure ?

→ Data structure is a way to store and org
data so that it can be used efficiently.

As per name indicates itself
organizing the data in memory.

The data structure is not
programming language like c, c++, Java et
It is set of algorithms that we can use
programming language to structure data in



Linear Data structure :-

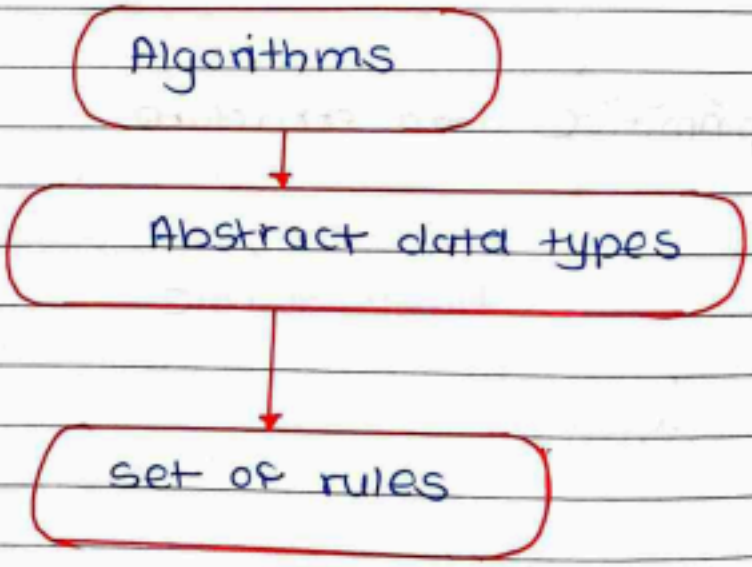
The arrangement of data in th
sequential manner is known as linear data
The data structure used for this purpose ar
Arrays, linked list, Stacks and queues.

In this data structures, one e
is connected to only one another element in

linear form.

Non-linear data structure :-
When one element is connec
to the 'n' number of elements known
non-linear data structures.
Example :- trees and graphs.
In this case, elements are
in a random manner.

Algorithms and Abstract Data types



Why →

To structure the data in memory, 'n' nu
of algorithms are proposed, and all th
algorithms are knowns as Abstract Da
Types.

An Abstract Data Type tells what is to be and data structure tells how is to be done

ADT gives us the blueprint while da structure provides the implementation part.

What is Data ?

Data can be defined as the eleme value / collection of values.
forexample :- student's name and its id e the data about student.

What is Record ?

Record can be defined as collection various data items
example :- student entity → name, address, course and marks can be grouped together to form record.

What is file ?

file is a collection of various reco of one type of entity
example :- if there are 60 employees in class then there will be 20 records in rela file where record contains info of em

What is Attribute and Entity ?

An entity represents class of certa objects. it contains various attributes. each attrib represents particular property of that entity.

What is need of data structures ?

As applications are getting complexed a amount of data is increasing day by da may arise following problems :-

Processor speed :- As data is growing d day to the billions of files per entity, p may fail to deal with that amount of da

Data Structure :- consider an inventory 106 items in store, if our application needs search for a particular item, it needs to 106 items every time, results in slowing d

multiple requests :- If thousands of us searching data simultaneously on a web ser then there are chances that to be failed during that process.

To solve this problems, d structures are used. Data is organized a data structure in a such way that a are not required to be searched and requir can be searched instantly.

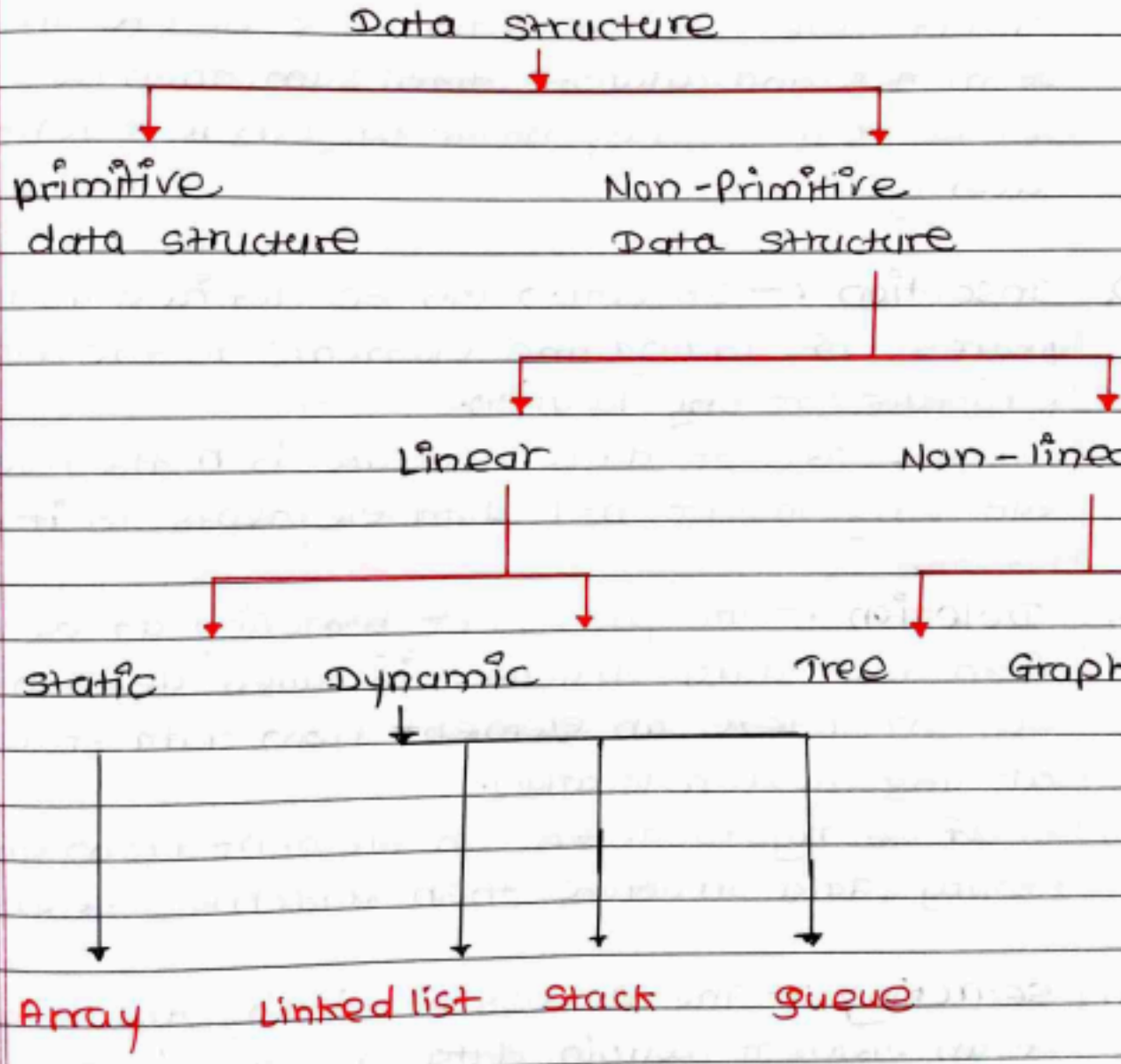
Advantages of data Structure :-

Efficiency :- If the choice of a data str for implementing a particular ADT is prop makes program very efficient in terms of space.

Reusability :- The data structure provides r means that multiple client programs can u data structure.

Abstraction :- The data structure specified by ADT also provides level of abstraction. The cannot see internal working of data structur it does not have to worry about implementa

* Data structure classification :-



Operations on data structure :-

1). Traversing :- Every data structure conta
set of data elements. Traversing data s
means visiting each element of data stru
in order to perform some specific operat
like searching or sorting.

Example :- If we need to calculate avera
of marks obtained by a student in 6 diff
subject, we need to traverse complete ar
of marks and calculate total sum, then we
devide that sum by no. of subjects ie. 6 to f
average.

2). Insertion :- Insertion can be defined as
process of adding the elements to the d
structure at any location.
If the size of data structure is n then u
can only insert n-1 data elements to it

3). Deletion :- The process of removing an ele
from the data structure is called deletion
we can delete an element from data stru
at any random location.
If we try to delete an element from a
empty data structure then underflow occu

4). Searching :- The process of finding the loc
of an element within data structure is call
searching. There are two algorithms to perf

searching, linear search and Binary search.

5). Sorting :- The process of arranging the data structure in a specific order is called as sor There are many algorithms that can be use perform sorting, for example, insertion sor selection sort, bubble sort etc.

6). merging :- When two lists list A and list B size M and N respectively, of similar type of elements, clubbed or joined to produce third list C of size (M+N), then this process is c merging.

DATA STRUCTURES AND ALGORITHM

What is Algorithm ?

An algorithm is a process or a set of required to perform calculations or some other problem-solving operations especially by a comp

It is not complete program or code ; it just a solution (logic) of a problem, which can represented either as an informal description u a flowchart or pseudocode.

characteristics of an algorithm.

Input :- An algorithm has some input values. can pass 0 or some input value to an algorithm.

Output :— We will get 1 | more output at ex
of an algorithm.

Unambiguity :— An algorithm should be unamb
which means that instruction in a
algorithm should be clear and simp

Finiteness :— An algorithm should have finite
means limited number of instructi

Effectiveness :— An algorithm should have finit
as each instruction in an algorithm
affects the overall process.

Approches in Algorithm :—

1). Brute force Algorithm :— The general logic str
is applied to design an algorithm. It is also
known as exhaustive search algorithm that
searches all possible to provide required solut

such algorithms have two types :—

1). optimizing
finding all solutions of a problem and then take out the best solution is known then it will terminate if the best solution is known.

2). sacrificing
As soon as t
best solution
found, then it
stop.

Divide and conquer :- This breaks down the algorithm to solve the problem in different methods. It allows you to break down problem into different methods, and valid output is produced for the valid input. This valid output passed to some other function.

Greedy algorithm :- It is an algorithm paradig that makes an optimal choice on each iterat with the hope of getting best solution. It is eas to implement and has faster execution time. B there are very rare cases in which it provides th optimal solution.

The major categories of algorithms are given b

Sort :- Algorithm developed for sorting the ite in a certain order.

search :- Algorithm developed for searching th items inside a data structure.

Delete :- Algorithm developed for deleting the existing element from the data structure.

Insert :- Algorithm developed for inserting an item inside a data structure.

Update :- Algorithm developed for updating the existing element inside a data structure.

Algorithm Analysis :-

The algorithm can be analyzed in two
i.e first is before creating the algorithm
second is after creating the algorithm.
There are two analysis of an algorithm.

Prior Analysis :-

Here, prior analysis is the theo
analysis of an algorithm which is done b
implementing the algorithm.

Posterior Analysis :-

Here, posterior analysis is a pr
analysis of an algorithm. The practical anal
is achieved by implementing algorithm us
any programming language.

Algorithm complexity :-

The performance of the algorithm
be measured in two factors :

Time complexity :-

The time complexity of an al
is the amount of time required to comp
the execution. The time complexity of an
is denoted by the big O notation.

Here big O notation is the a
-tic notation to represent time comple
The time complexity is mainly calculated
counting the number of steps to finish
execution.

```
sum = 0 ;
// suppose we have to calculate the sum of n
  numbers.
for i = 1 to n
sum = sum + i ;
// when the loop ends then sum holds the
of n numbers.
return sum;
```

In above code, the time complex
of the loop statement will be atleast n, an
if value of n increases, then time complexit
also increases.

We generally consider the worst
complexity as it is maximum time taken for
any given input size.

Space complexity :—

An algorithm's space complexity is
the amount of space required to solve a pr
and produce an output. Similar to the time
complexity, space complexity is also expressed
big O notation.

Space complexity = Auxiliary space + Input size

The following are the types of algorithms :

Search Algorithm :-

on each day, we search for some in our day to day life.

similarly, with the case of com huge data is stored in a computer that whenever user asks for any data then th computer searches for that data in the memory and provides that data to the us There are mainly two techniques available t search data in an array :

- Linear search
- Binary search

Sorting Algorithms :-

sorting algorithms are used to rea elements in an array or a given data struc either in an ascending or deseending order The comparison operator decides the new or of the elements :

Asymptotic Analysis :-

The time required by an algorithm com
under three types :

Worst case :- It defines the input for which
the algorithm takes a huge time.

Average case :- It takes average time for
the program execution.

Best case :- It defines the input for which
the algorithm takes the lowest time.

Asymptotic Notations :-

The commonly used asymptotic nota
used for calculating the running time complexi
of an algorithm is given below :

1). Big oh notation (O) :-

This measures the performance of a
algorithm by simply providing the order of
growth of the function.

This notation provides an upper bou
on a function which ensures that function ne
grows faster than the upper bound.