June 23, 2025

Unit: 1

Introduction to Data Structures

* The data structure is describe the way data is stored and the term algorithm is used to describe the way data is processed

Algorithm + DS = Program

* Data type & data object
  - ↳ Data refers to value or valueset, data is divided into atomic or non-atomic data

- Atomic data
  - ↳ It is a single or non-decomposible entity for e.g., integer.

- Non-Atomic dat
  - ↳ A data that can be broken into sub-fields that have meaning is called composite data. for e.g., data: DD/MM/YYYY

June 24, 202

classmate
Date
Page

Data types in C

- User defined -> Structure, union, enum
- Built-in -> int, char, float, double, void, etc
- Derived -> Arrays, functions, pointers

* User-defined datatype

① Structure

↳ It is a tool for packing together logically related data items of different types (Heterogeneous)

for e.g.,

```
struct employee {
    char empname[20];
    int empid;
    float basic_sal;
};
```

② Union

↳ Unions are very much similar to structure where both are used to group no. of different elements, only the difference is the member of union share same storage area eventhough the individual elements ~~may~~ different in types.

for e.g.;

```
union id{
    char color[12];
    int size;
};
```

③ Enum
↳ Enumeration type consist of integer constants
for e.g.;
enum demo {const1, const2, ..., const3}

* Derived datatype

① Array
↳ Assigns single name to whole group of elements

② Pointer
↳ Pointer is a memory location that hold memory address.

③ Data object
↳ Data object represent container for data values (a place where data values maybe stored and later retrived). Data object is a set of attributes. for e.g., data object alphabet is defined D = {A, B, C, ---, Z}.

④ Abstract datatype (ADT)

↳ An abstract datatype (ADT) is a data declaration package together with the operation that are meaningful for the datatype.

ADT includes:

① Domain: Collection of data
② Function: Set of operation on the data
③ Axioms: Set of axioms or rules of behaviour governing the interaction of operations (protocol).

Advantages of ADT

* Encapsulation
  ↳ Information hiding
  ↳ Implimentation detail hiding.
  ↳ A.D.T. is reusable DS
  ↳ Encapsulation ensures that data cannot be corrupted

June 25, 2025

Data Structure: The Beginning

Data structure (DS) is a set of domains D, a design domain $\in D$, a set of function F & set of axioms A. The triple (D, F, A) denotes the data structure 'd' and it'll usually written as 'd', where domain (D) denoted data object, function (F) denotes set of operation that can be carried on the data objects and axioms (A) denotes the property and rules of operation.

Read
* Operation of DS
1. Insertion.
2. Deletion.
3. Searching.
4. Traversing.
5. Sorting.
6. Merging.

* Need of DS
↳ DS is used to study how data is stored in a computer so that operations can be implemented efficiently. DS is specially important when you have large amount of information. It is also use to organize data or efficient storage & manipulation. DS helps us to understand the relationship of one element with other.

most imp
* Advantages of DS

① It allows information stored on disk very efficiently.

② We can access data anytime & anywhere

③ It allows processing of data on software system.

④ It is secure way of storing data

most imp

* Types of DS /

DS can be divided in two types

Data Structure

- Linear
  - ↳ Array
  - ↳ Linklist
  - ↳ Stack
  - ↳ Queue
- Non-Linear
  - Tree ↵
  - Graph ↵

2/4 marks

* Linear DS

↳ ① In linear DS, data elements are arranged in linear or sequential order

② In this DS, single level is involve.

③ Data item can be traverse in single run.

④ It is easy to implement, but, memory is not utilize inefficient way.

* Non-linear DS

↳ ① A DS is said to be non-linear, if element are attached in hierarchy manner & multiple level are involve in this type of DS elements do not form a sequence

② Data items cannot be traverse in single run.
③ These are difficult of implement but memory is utilized in efficient way.

exam ★ Algorithm analysis

↳ Algo. is finite sequence of instruction, each of which have clear meaning & can be executed with finite amt. of time.

★★★ Characteristic of algo

① Input: Algo should accept zero or more input

② Output: Algo must produce atleast one result / o/p

③ Definiteness: Each step of algo must be clear & meaningful

④ Finiteness: Algo must halt, it must terminate after finite step

⑤ Efficientness: Instruction can be execute in finiteness time

June 26/25

Imp

June 26/2025

classmate
Date
Page

Imp Performance Analysis

↳ It is an algo in which process of calculating space require by that algo. performance analysis uses following measures

① Space req. to complete the task of that algo. (Space complexity)

② Time req. to complete the task of that algo (time complexity)

Space Complexity

↳ Total amt. of computer memory req. by an algo to complete its execution is called as space complexity of that algo.

There are 3 reasons

① Instruction space: it is amt. of memory used to store compiled version of instruction.

② Environmental stack: It is the amt. of memory used to store information of partially executed func^n at the time of func^n call

③ Data space: It is amt. of memory used to store all the variable and constant.

* Time Complexity

↳ Time complexity of an algo. is the total amt. of time required by an algo to complete its execution.

* Frequency count
  ↳ It tells how many time the statement is executed in program.

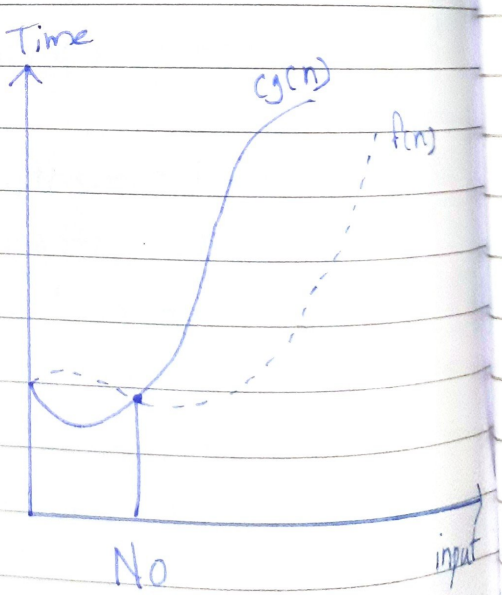
  * Time req. by each statement depends on,
    ① Time req. for executing it once
    ② No. of time the statement is executed

* Asymptotic Notation
  ↳ Asymptotic notation of an algo is a mathematical representation of its complexity

Type of asymtotic notation

① Big oh (O) notation

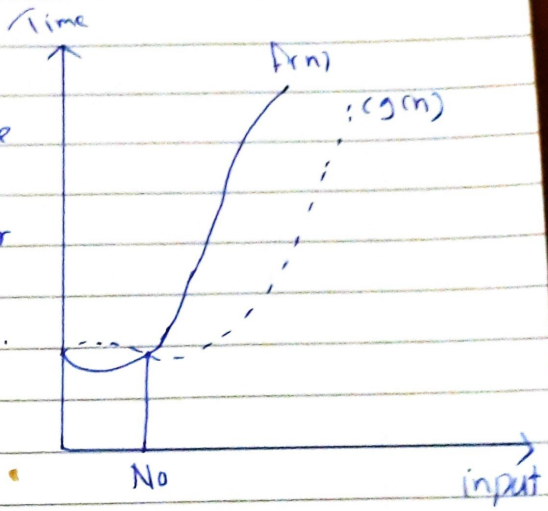
Big oh (O) notation is used to define the upper bound on an algo. in terms of time complexity. This notation indicates the max. time req. by an algo for all input value that means Big oh (O) notation describe the worst case of algo. time complexity. It is defined as follows



$$f(n) <= cg(n)$$
$$\forall\ n >= no\ \&$$
$$c > 0$$
$$no >= 1$$

② Big omega (Ω) notation

It is used to define the lower bound of an algo. in terms of time complexity. It always indicates the minimum time req. by an algo. for all input values. That means Big omega (Ω) describe best case of algo. time complexity.



$f(n) >= cg(n)$

$\forall\ n >= n_0$ &

$c > 0$

$n_0 >= 1$

End of Chapter