

Q. What do you understand by data structure? (Elaborate
are important? List out its application areas.

⇒ A data structure is specialized format for organizing, processing, retrieving & storing data. Data structure is defined as the group of data elements which provides an efficient way of storing & organising data in the computer so that it can be used efficiently.

Data structures have many applications, are -

- Data Storage

Data structures facilitate efficient data persistence, like specifying attribute collections & corresponding structure used in database management system to store records.

- Data Exchange

Organized information, defined by data structure can be shared b/w applications like TCP/IP packets.

- Resource & Service Management

Data structure as linked lists can enable core operating systems resource & service to perform function like file directory management & processing scheduling queues.

- Scalability

Big data application typically rely on data structures to manage & allocate data storage across many distributed storage locations.

Advantage of Data structure.

- Data structure facilitate effective data storage device of data
- The use of structure make it easier to retrieve data from a storage device
- The data structure allows for the effective & efficient of both little & big amounts of data.
- Manipulation of vast amounts of data is simple when a proper data structure technique is used.
- The use of good data structure may assist a programmer to save a lot of time or processing time while performing tasks such as data storage retrieved

Linear Data Structure

Non-linear Data Structure

- Data elements are sequentially connected & each element is traversable through a single run.
- All data elements are present at a single level.
- Linear data structure is easier to implement.
- They are not very memory friendly & are not utilizing memory efficiently.
- e.g.: Array, List, Queue, Stack etc.
- Data elements are hierarchically connected & are present at various levels.
- Data elements are present at multiple levels.
- They are difficult to understand.
- Non-linear data structure uses memory very efficiently.
- e.g.: Graph, Map, Tree.

Primitive Data Structure

Non-primitive Data str.

- It is a kind of data structure that stores the data of only one type.
- Primitive will contain some value i.e. it cannot be NULL or NULL value.
- The size depends on the type of data structure.
- It starts with a lowercase character.
- e.g.: integer, character, float.
- It is a type of data str. that can store the data of more than one type.
- Non-primitive can consist of NULL or NULL value.
- The size is not fixed.
- It starts with an uppercase character.
- e.g.: Array, stack, linked list.

Q. Operation on different Data structure

- Traversing

Traversing Data structure means to visit the element stored in it. It visit data in a systematic manner. This can be done with any type of DS.

- Searching

Searching means to find a particular element in the given data structure. It is considered as successful when the required element is found. searching is operation which can performed on data-structure like array, tree, graph etc.

- Insertion

It is the operation which we apply on all the data-structure. Insertion means to add an element in the given data structure. The operation of insertion is successful when the req. element is added to the required data-structure.

- Deletion

It the operation which we apply on all data-structure. Deletion mean to delete an element in the given structure. The operation of deletion is successful when the required element is deleted from data structure.

Q. What is ADT? How it differ from data type?
What are the benefits of using ADT.

⇒ ADT is defined by set of value & a set of operation. Definition of ADT only mentions what operation are to be performed but not how these operation will be implemented.

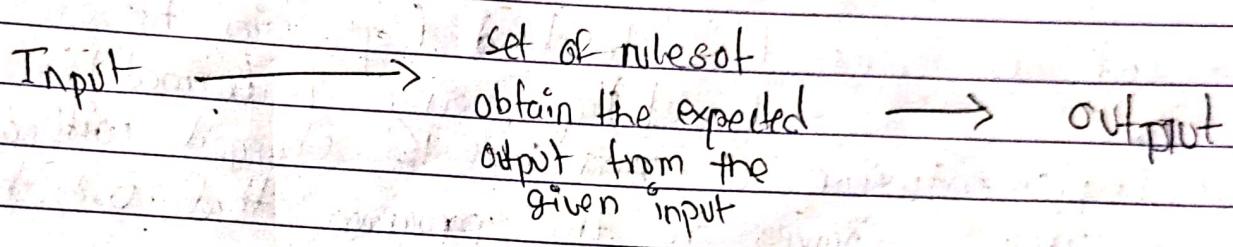
Benefit of using ADT:-

- Code is easier to understand (e.g.: easier to see high-level step being performed)
- Implementation of ADT can be changed without requiring changes to the program that uses the ADT
- ADTs can be reused in future programs.

⇒ ADT is more of logical description, while a Data structure is concrete. This means data structure allows us to look at the representation whereas the abstract one doesn't. ADT can be implemented in high concept but data structure can only be implemented in simple one. ADT uses "class" & data structure uses "structure".

Q What is algorithm? What are the characteristics of good algorithm.

→ Algorithm is a step-by-step procedure that define a set of instructions that must be carried out in a specific order to produce the desired result. Algorithms are generally developed independently of underlying languages, which means that an algorithm can be implemented in more than one programming language.



Algorithm.

Characteristics of good algorithm.

- **Input:** Algorithm requires some input value. An algorithm can be given a value other than a as input.
- **Output:** At the end of an algorithm, you will have one or more outcome.
- **Unambiguity:** A perfect algorithm is defined as unambiguity, which means that its instructions should be clear & straight forward.

~~Pages~~

liteness :- Finiteness in this context means that the algorithm should have limited no. of instruction ie: the instruction should be countable.

Effectiveness :- Because each instruction in an algorithm affects the overall process, it should be adequate.

Language independence :- An algorithm must be language-independent, which means that its instruction can be implemented in any language & produce the same result.

What are asymptotic notations? Explain its type.

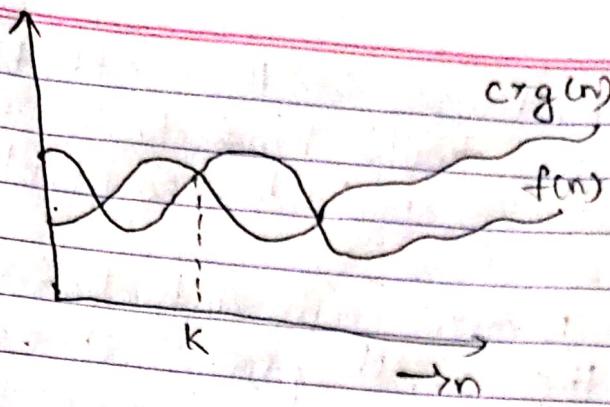
Asymptotic notations are the mathematical notations used to describe the running time of algorithm when the inputs tends towards a particular value or a limiting value.

Eg:- in bubble sort, when the input array is already sorted, the time taken by the algorithm is linear ie. the best case

Its type :-

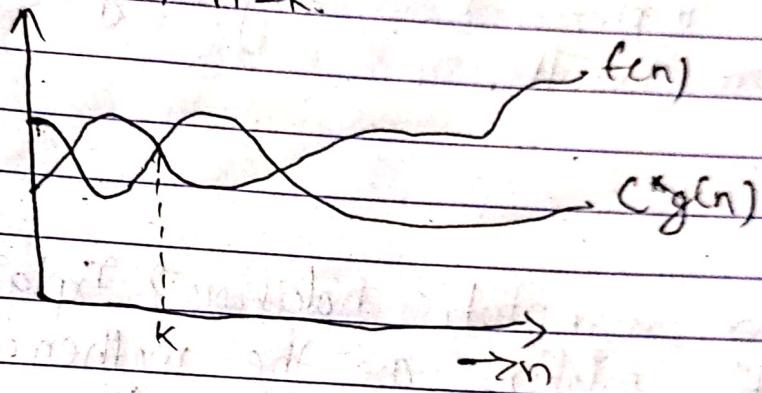
Big Oh notation

The function $f(n) = O(g(n))$, if & only if there exist positive constant C & k such that $f(n) \leq C * g(n)$ for all $n \geq k$



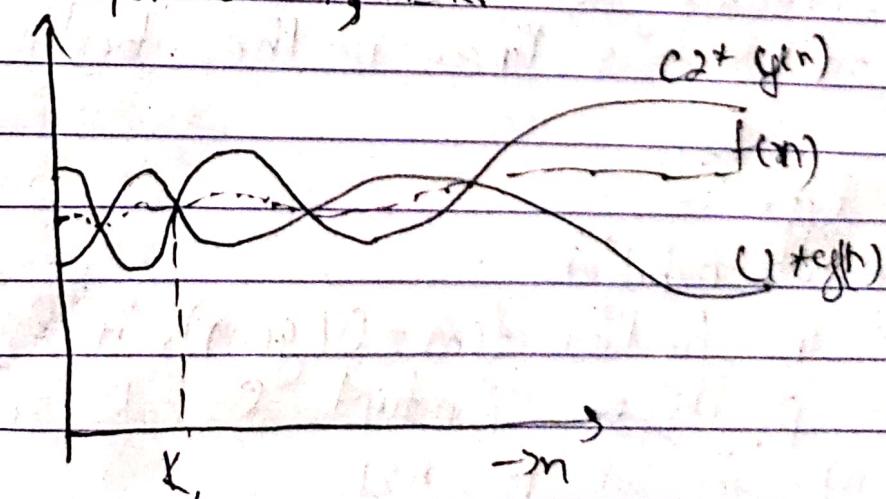
(ii) Big Omega notation:

The function $f(n) = \Omega(g(n))$, if & only if there exist a positive constant c & k such that $f(n) \geq c^* g(n)$ for all $n, n \geq k$.



(iii) Big Theta notation:

The function $f(n) = \Theta(g(n))$, if & only if there exist a positive const. c_1, c_2 & k such that $c_1^* g(n) \leq f(n) \leq c_2^* g(n)$ for all $n, n \geq k$.



Q. How can you say stack is an ADT? Explain

→ Stack is abstract data type because it's hide how it is implemented like using or linked lists. A real-world stack allows operations at one end only for example we can place or remove a card or plate from the top of the stack only. Likewise stack ADT allows all data operations at one end only. At any given time, we can only access the top element of a stack.

Q Point out limitations & advantages of diff asymptotic notations.

→ advantage

1. To perform analysis of algorithm, complexity is required but when we calculate the complexity, the exact amount of resources is unknown which can be known using asymptotic notation.
2. The efficiency of the algorithm can be measured.
3. Asymptotic analysis can be done which helps us to study the change in performance of algorithm with the change in order of input size.
4. It become general & not programming language specific.

limitations.

1. Some algorithms might be too diff. to analyze mathematically.
2. Smaller input size are neglected which might dominate running time.
3. Sufficient information may not be available to calculate the behaviour of algorithm.

Contd—

(Page)

Operations of data structure are

④ Updating

→ It update or modifies the data in data structure.
New data may be stored or previous data may be deleted.

⑤ Sorting

→ It is the process of arranging all the data items in a data structure in particular order (ascending or descending).

⑥ Merging

→ It is the process of combining the data items of two diff sorted list in a single sorted list