

Arrays and Strings(DSA Made easy)

① Data types-

Integer - 2 bytes

float - 4 bytes

At top level there are two types of data types-

① System defined data types (called Primitive data types)

② User defined data types (~~non primitive data types~~)

① Primitive data types-

int, float, double, bool

The number of bits (or memory allocation) depends on the programming language, compiler, and operating system.

② User defined data types - When the system defined data types are not enough, then most programming language allows the user to define their own datatype. This is called user defined data types.

```

struct newType {
    int data1;
    float data2;
    char data3;
}

```

① What is data structure:

→ So basically when we have data in arrays, we need some mechanism for manipulating that data to solve problems.

Data structures is a particular way of storing / organising data in computer memory so that it can be used efficiently.

→ Depending on how the elements are organised data structures are classified into two types

① Linear data structure

② Non linear

elements are accessed in sequential order

(but it's not compulsory to store them sequentially)

e.g. linked list -
stacks/ queues

Tree, graph -

② Abstract data types (ADT)

→ We all know primitive data types support basic operation like addition, multiplication etc. The system provides the implementation for the primitive datatypes. So for user defined data types also we need to define operations to implement its implementation. In user defined data types are defined along with operations.

So to simplify the process of solving problem, we concern data structure

with these operations, and we call them abstract data structures (ADT).

→ ADT has two things

→ ① data declaration

② declaration of operation.

Algorithm →

Formal definition of algorithm -

The step by step process of solving a problem.
In traditional study there are two main criteria for helping the merits of algorithms.

→ Correctness → does the algorithm give solution

→ efficiency → run a finite number of steps.

↳ how much resources in terms of memory and time does it takes.

Why analysis of algorithm?

→ We can solve a problem in different ways but we need to select the most efficient way. Similarly in computer science also we can solve a specific problem by different algorithms.

So algorithm analysis helps us to determine which algorithm is most efficient to solve a problem.

① What is Running time complexity.

→ It is the process of determining how processing time increases as the size of the problem (input size) increases.

② How to compare algorithms -

→ The order of solution of comparing algorithm as a function of the input size and how the algorithm behaves based on the ~~on~~ input size.

Types of analysis of algorithm.

→ Best case -

→ defines the input for which the algorithm takes least time.

→ Input is the one for which the algorithm runs fast.

→ Worst case -

→ defines the input for which the algorithm runs slow.

→ Input is the one for which the algorithm runs slow.

→ Average case -

Lower bound \leq Average Time \leq Upper bound.

$$f(b) = O(gcn)$$

$$f(n) = n^k + 100n^2 + 10n + 50$$

$$\begin{aligned} f(n) &= n^k(n^r + 100) + 10n + 50 \\ &= n^r \end{aligned}$$

Big O notation

→ Linear scaling

Big $O(n)$

→ Constant Big $O(1)$

→ Asymptotic notation -

— o —

ARRAYS

Arrays are abstract data types /

Abstraction \rightarrow knowing the details.

→ Array is a collection of elements accessible by index.

→ Store data contiguous memory.

→ Faster data retrieval and deletion,
but slower when up data. Since array
has a fixed size.

Addition and insertion is costly.

→ Static and dynamic arrays-

→ Static - size cannot be changed

→ Dynamic \rightarrow size can be changed -