Data Structur

Defection

Systematic way to organise data so that it can be used efficiently

Abstract Data Type (ADT) is a type (or class) for objects whose behavior is defined by a set of values and set of operations

Group of data stored togather under one name
-Instead of creating multiple variables of same type.
-Storing strings is equivalent of storing a sequence of characters.
-Ex: array

The Challenge is to select the appropriate data structure for the problem

Classification

Based on Type

Primitve

Storing the value of **only one** data type

Ex:
Float
Integer
Boolean
Character

Non-Primitive

Storing the value of **more than one** data type

Ex:
List
Tuple
Array
Stack
Queue
Dictionary

Based on Relationship

Linear

Data elements **are arranged sequentially or linearly**

Single level is involved.
It can traverse all the elements in single run only.

Easy to implement because computer memory is arranged in a linear way.

Every element is **attached to its previous and next adjacent**

Ex: Array , Stack , Queue , linked list

Non Linear

Data elements **are not arranged sequentially or linearly.**

Hierarchical **multiple levels** are involved
It can't traverse all the elements in single run only.

implementation is complex.

It utilizes computer memory **efficiently in comparison to a linear data structure.**

Ex: Tree , graph

Based on Memory Allocation

Static

Collection of data in memory which **have a fixed size**

It can **store a limited amount of elements** or data in memory.

Ex: Array

Dynamic

Collection of data in memory which do **not have a fixed size at**

It can **store a variable amount of elements** or data in memory.

Its **size can be modified** during the operations performed on it

Ex: linked list , Stack , Queue

Based on level of abstraction

physical

Actual implementations of data structures in memory.

Ex: Array , linked list

Logical

Logical implementations of data over physical memory.

Ex: Stack , Queue , tree

Operations

Different types of operations can be performed for the manipulation of data in every data structure.

Traversal

accessing each record/node exactly once so that certain items in the record may be processed.

Search

Finding the location of the desired node with a given key value, or to satisfy one or more conditions.

Insertion

Adding a new node/record to the structure.

Deletion

Removing a node/record from the structure.

Sort

arranging all data items in a particular order.

Merge

combining the data items of two different sorted list into a single sorted list.

How To Choose Data Structer

Based on the **operations** that will be performed

A **linked list** used to perform **insertions and deletions.**
An **array** is used to perform **indexing.**

Based on the **time complexity** of the operations

binary search tree used to perform **searches frequently**

Based on the **space complexity** of the operations

Array used to **store a lot of data**

Based on the **amount of memory** used

Linked list used to **store a lot of data**

Presented with xmind