**Data structure** is a way to store , organize, manage data in a computer,

so that it can be used efficiently.

it helps programmers to acceess/search, insertion,deletion, modification

just like physical tools like bag,shelves,drawers.

**(Q)Why are data structure important?**

Different problems require different data structure for-

**1.speed(Time efficiency)**

**2.space(Memory efficieny)**

**3.Simplicity(Easy to implement,read,maintain)**

**For fast lookup** -> use set,dict

**For ordered data** -> use list, sorted containers

**For minimum/maximum retrieval** -> use heap

**For relational structure** -> use tree

Types of data structure:

**1.Linear Data structure(These store data in sequence):**

Array/List, Stack ,Tuple,

Queue(Simple,Circular,Priority, Deque(Double-Ended Queue) )

Linked List(Singly, Doubly, Circular)

**2.Tree based Data Structure(These store data hierarchically):**

Binary Tree, Binary Serach Tree,AVL tree, Red-Black tree, Segment tree,Fenvic tree,Trie tree, Heap(Min-heap, Max-heap)

**3.Hash-Based Data structure(use Hashing for fast access):**

Hash Map/Hash Table(dict in Python), Hash Set(set in python), Counter(use dict)

**4.Graph Based Structures(For representing networks and relationships):**

Graph(Adjacency list, Adjacency matrix)

Weighted Graph/Unweighted Graph

Directed /Undirected Graph

Cyclic/Acyclic Graph

**5.Specialized Structures(Use for advanced or specific use cases.):**

Disjoint Set/Union-Find ,Bloom Filter, Skip List, LRU Cache, Suffix Tree/Array

B-Tree/B+Tree (uses in databases), Deque (Double-Ended Queue)

A Graph is a pair G=(V,E)

where: V= set of vertices(nodes) E = set of edges(connections between nodes)

6



4 5



1



3 2



**1.2Directed vs Undirected**

**2.Weighted vs Unweighted**

**3.Cyclic vs Acylic**

Graph can be represented as: **(a)Adjacency List, (b)Adjacency Matrix**

**(a)Adjacency List Representation:**

For a Directed graph: Total list of length=[E]

For an undirected graph: Total list length = 2[E]

Space Complexity : Θ(V+E)

Used in most graph algorithms due to memory efficiency.