**Data Structure**

**1. Time Complexity:**

Big O Notation

**2. Linear VS Non-Linear**

**Linear:**

**One after one**

1. Array,

2. Linked list,

3. Stack

4. Queue

5. Hashing

Non-Linear:

1. Tree

2. Graph

3. Heap

4. Trie

5. Set

**Array:**

Array is a linear data structure. Its kind like the list but the biggest difference Is an Array can store only same data type and array sizes are fixed. When declares any array the sizes of array need to declare.

Disadvantage:

1. Fixed size (memory waste)
2. Homogenous (lack of flexibility)

For avoiding homogenous array or normal array I can use referential array.

Referential array is kinda different from normal array because it A **Referential Array** is an array that stores references to objects or other data structures rather than the actual data itself. In programming, particularly in object-oriented languages like Java, Python, or C#, referential arrays are widely used to hold references to instances of classes or other complex data types.

For example:

12.4

123

anis

1001 1002 1003

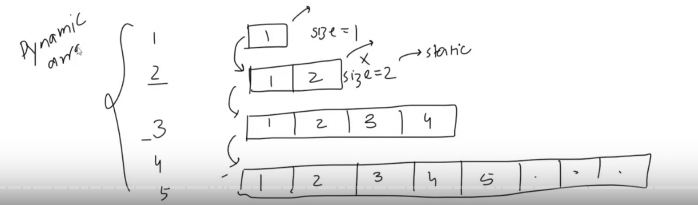
1001 1002 1003

Disadvantage:

1. This are slower than normal array because of when a value is search from the array usually in normal array the value is directly search from the array in linear way by using address. But in Referential Array when any value Is search from the array it first search from the address number then the address pointer the value address which are attach with the value. Then the expected value can be find.

The major issue for static array is fixed size array.

So, the solution is dynamic array. Dynamic array its actually depend on the concept there are doesn’t any special keyword



Dynamic array is not fixed array. The size is increase as the given input. The biggest similarity is in Python Language List are the **Dynamic Array.**

**Linked List**

Linked list is the Linear Data Structure. Linked list is basically a collection of nodes.

Data

Address

500

Data

Address

500

Data

Address

700

Actual Address 400 Actual Address 700

1st Node are called **Head**

Last Node are called **Teal**

Last Node can be identify by the Null address.

**\*\*\*\*\*What is the actual benefit of linked list other than Array?**

**Ans: Linked list is actually uses for solving the weakness of array.**In Array, when we do write operation (write operation means insert and delete) linked list follow the time complexity of O(n). That means when the operation is increased time would be increased. That’s mean when we apply for inserting a number on a Big Number of List. It swapping each of the number. That is so much time consuming.

**In case of Linked List,** When we apply write operation linked list follow O(1) that are constant. That’s mean when the operation applied. Time will be same it doesn’t depend on the data.

**For making a individual nodes in manually:**

**Two variable need 1 is data and other is address of the next node.**

**The first linked list node will be empty. The node are called head that next address will be none.**

**Because no other node are linked with that.**

**In Linked List there are four main operation:**

1. **Insert**
2. **Traverse**
3. **Delete**
4. **Searching**

**Insert:**

1. **Head**
2. **Tail (append function)**
3. **Mid insert (insert function)**

**Traverse:**

1. **Print (str method)**

**Delete:**

1. **Head**
2. **Tail (pop method)**
3. **Value (remove method)**
4. **Index (find method)**

**Searching:**

1. **Value**
2. **Index**