**LINK LIST AND MATRIX**

Ways to maintain a list in a memory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |

**1** **Array**

|  |  |
| --- | --- |
| data | address |

|  |  |
| --- | --- |
| data | address |

**2 Linked List**

Type of Linked list

1 **Single linked list:** *Navigation is forward only (node has two parts: data-actual data, link-contains the address of the next node of the list)*

*Example: suppose we want to store a list of numbers: 23, 54, 78, 90*

|  |  |
| --- | --- |
| *23* | *2000* |

|  |  |
| --- | --- |
| *54* | *3000* |

|  |  |
| --- | --- |
| *78* | *4000* |

|  |  |
| --- | --- |
| *90* | *Null* |

1000 2000 3000 4000

*Head 1000 - pointer*

2 **Doubly linked list**: *Forward and backward navigation is possible*

**3 Circular linked list**: *Last element is linked to the first element*

*Operations:*

***1Traversal Operation***

*1Start with the head of the linked list*

*Access the data if head is not null*

*2 Go to the Next Node Access node data*

*Continue until last node*

***2 Insertion***

***3 Deletion***