|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Structure | Important | Advantages | Disadvantages | Application | Complexity |  |  |  |  |
| Singly Linked List or One way chain | -most used data structure | 1. Dynamic size 2. Ease of insertion / deletion | 1. Random access is not allowed 2. Extra memory space for a pointer 3. Not cache friendly 4. Not good for modifying random elements | 1. Implement stack and queue 2. To prevent collision in hash map | Worst  O(n)-Access  O(n)-Search  O(1)-Insertion  O(1)-Deletion  O(n)-Space Com |  |  |  |  |
| Doubly Linked List |  | 1. Reversing is easy 2. Deletion is easy 3. Dynamic size | 1. Extra memory due to two pointers 2. Elements can only be accessed sequentially 3. Random access is not allowed 4. Operations require more time (handling extra pointers) | 1. Undo redo in apps 2. Used in navigation system. 3. Deck of cards game 4. Previous next in apps 5. Thread scheduling (moving thread from one queue to other) 6. Implement Binary tree, hash table, stack |  |  |  |  |  |
| Circularly Linked List |  |  |  | 1. Time sharing mechanism for multiple users 2. Round Robin 3. Multi-player games 4. Undo in apps |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Linked List-most used data structure