

Data Structure

22/5/24

Name- Atlan Santosh Joseph

Roll no.- 230101004

• Time complexity

| <u>Dynamic Array</u> | <u>Linked list</u> |
|----------------------------------|-------------------------------------|
| • <u>resize()</u> $O(N)$ | • <u>getsize()</u> $O(1)$ |
| • <u>getsize()</u> $O(1)$ | • <u>isEmpty()</u> $O(1)$ |
| • <u>isEmpty()</u> $O(1)$ | • <u>append at list()</u> $O(N)$ |
| • <u>insert()</u> $O(N)$ | • <u>add first()</u> $O(1)$ |
| • <u>remove()</u> $O(N)$ | • <u>insert Add()</u> $O(N)$ |
| • <u>append()</u> $O(1)$ | • <u>delat()</u> $O(N)$ |
| • <u>prepend()</u> $O(N)$ | • <u>middle()</u> $O(N)$ |
| • <u>rotate()</u> $O(N)$ | • <u>reverse()</u> $O(N)$ |
| • <u>reverse()</u> $O(N)$ | • <u>index at()</u> $O(N)$ |
| • <u>middle()</u> $O(1)$ | • <u>merge two list()</u> $O(N)$ |
| • <u>first()</u> $O(N)$ | • <u>inter leave()</u> $O(N)$ |
| • <u>merge()</u> $O(N)$ | • <u>two parts()</u> $O(N)$ |
| • <u>inter leave()</u> $O(N)$ | • <u>rotate()</u> $O(N)$ |

Advantages & disadvantages :-

⇒ linked list

Adv

1. Dynamic Size : We can easily change the size of linked list by allocating, deallocating memory as needed.
2. No wastage of space : No pre-allocating of memory required.

disadv

1. Sequential access : We can not access any element of linked list directly
2. Non-contiguous memory allocation

⇒ Dynamic Array

Adv

1. Random access : We can directly access any member of the array
2. contiguous memory

disadv

1. Wastage of space : If not inserting enough elements as per pre-allocated memory
2. Inserting element at middle requires shifting.