

Linked list and Dynamic Array

1. **Less space occupied in dynamic array , linked list have data as well as address always double space while dynamic array have double occupied space for few interval of time**
2. **Insertion , deletion** generally $O(n)$ in dynamic array plus shifting other is also a headache while $O(n)$ in Linked list but less painful
3. **Reversing** same $O(n)$
4. **Random access** $O(n)$ in linked list(need to iterate over each element till we get) , $O(1)$ in dynamic array
5. **Merging** $O(1)$ in linked List $O(n)$ in dynamic array
6. **Reverse** $O(n)$ for both
7. **K_rotation** $O(k)$ in linked list $O(n)$ dynamic array
8. Arrays need **contiguous space** in memory while linked list nodes can be at arbitrary positions.

Both have its own advantages and disadvantages as well no one competes both have its own value and use case.

If my task requires continuous insertion and deletion and has no continuous space in memory then we use a linked list.

If task required constant random access based on index or its position then certainly we will use array

Other use case of linked list is that it is used to implement many data structures like stack , queue