**Analysis of Linked Lists**

**Time Complexity:**  
• Access by index – O(n) (need to traverse)  
• Insertion at beginning – O(1)  
• Insertion at end (with tail pointer) – O(1)  
• Insertion at middle – O(n) (find the position first)  
• Deletion at beginning – O(1)  
• Deletion at end or middle – O(n) (need to traverse)

**Space Complexity:**  
• Each node stores data + one or two pointers  
• Singly linked list uses less space than doubly linked list  
• Slightly more memory than arrays due to node pointers

**Advantages:**  
• Dynamic size – no need to predefine length  
• Efficient insertions/deletions compared to arrays  
• Better for applications where frequent modifications happen

**Limitations:**  
• Slower access – must traverse from head  
• Extra memory for pointers  
• More complex to implement than arrays