Linked List Study Note

Name：黃以蔡

Student ID：1133313

Date: 10 / 10 Linked List Study Note

1. Definition

A Linked List is a liner data structure composed of nodes, where each node contains data and a pointer to the next node.

It sloves the problem of low efficecy in array insertion and deletion, supporting dynamic size.

2. Visualization

Head → [10 | next] → [20 | next] → [30 | next]

→ [40 | next] → [50 | next] → [60 | next] → NULL

- **Each box** represents a node containing data and a pointer
- Head points to the first node
- The last node's Next points to NULL

※ Doubly linked lists have a Prew pointer pointing to the previous node.

3. charaterristics

- ordering : Ordered, nodes are linked sequentially
- Indexing : doesn't support O(1) random access, must traverse Head (O(n))
- Dynamic size : Can grow or shrink dynamically
- Memory layout : Non-contiguous, nodes can be scattered in memory
- Typical operations : Insert, delete, search, traverse, link node.

4. Time / Space Complexity

Operation	static Array	Dynamic Array
Access / search	O(n)	O(n)
Insert	O(1)	O(1)
Delete	O(n)	O(n)
Extra memory	O(n)	O(>n)

5. Limitations

- Can't access elements directly by index, traversal is required
- Requires extra memory for pointer
- slow for random access

Date: /

6. Pros / Cons

Pros :

- High efficiency for insertion and deletion
- Dynamic size, memory usage is flexible
- Can implement stack, queue, graph adjacency list

Cons :

- Accessing a specific element requires O(n) traversal
- Pointer management can be error-prone
- Higher memory overhead

7. Use Cases

1. Dynamic data collections
   - e.g. task lists where elements are frequently added/removed
2. Implement stack or queue
   - avoid array shifting overhead
3. Graph adjacency lists
   - store neighbors of each node