Made By:
Kenan Gazwan

Linked Lists

ICS202-Summary

King Fahd University of Petroleum and Minerals

Telegram: @KenanGazwan

(Linked Lists)

✓ Singly Linked Lists

A dynamic data structure consisting of a sequence of nodes, forming a linear ordering.

It has nodes + (tail & head) references.

Each Node has:

→ info

→ next (reference to the next node)

Singly Linked List Operations Complexity

Operation	Add To Head	Add To Tail	Delete From Head	Delete From Tail	Search/Traversal
SLL	*O(1)*	*O(1)*	*O(1)*	***O(1)*** *has tail ref* ***O(n)*** *has no tail ref*	*O(n)*

Singly linked lists vs. 1D-arrays

ID-array	Singly Linked List
Fixed size: Resizing is expensive	Dynamic size
Insertions and Deletions are inefficient: Elements are usually shifted	Insertions and Deletions are efficient: No shifting
Random access i.e., efficient indexing	No random access
No memory waste if the array is full; otherwise, may result in much memory waste.	Extra storage needed for references; however, uses exactly as much memory as it needs
Sequential access is fast. [Memory locations are contiguous]	Sequential access is slow. [Memory locations are not contiguous]

Time Complexity: Singly Linked Lists vs. 1D-arrays

Operation	Insert beginning	Insert end	Insert middle	Delete beginning	Delete end	Delete middle	Search
1D-Array	*O(n)*	*O(1)*	***O(n)*** *due to shifting*	*O(n)*	*O(1)*	***O(n)*** *due to shifting*	***O(n)*** *Linear Search* ***O(logn)*** *Binary Search*
SLL	*O(1)*	***O(1)*** *has tail ref* ***O(n)*** *has no tail ref*	*O(n)*	*O(1)*	*O(n)*	***O(n)*** *due to search*	*O(n)*

✓ Doubly Linked Lists

A dynamic data structure consisting of a sequence of nodes, forming a linear ordering.

It has nodes + (tail & head) references.

Each Node has:

→ info
→ next (reference to the next node)
→ previous (reference to the previous node)

Doubly Linked List Operations Complexity (vs Singly Linked List)

SLL and DLL have the same complexity operations except: **deleteFromTail (delete end)**: -

- ✓ **SLL = $O(n)$**
- ✓ **DLL = $O(1)$**

✓ Circular Linked Lists

A sequence of nodes in which every node has a link to its next node in the sequence and the last node has a link to the first node.

It has nodes + last reference.

Each Node has:

→ info
→ next (reference to the next node)

Circular Singly Linked List Operations Complexity

CSLL and SLL have exactly the same complexity operations.

Complexity Summary:

DLL Operations = SLL Operations (Except deleteFromTail):

- **SLL = $O(n)$**
- **DLL = $O(1)$**

CSLL Operations = SLL Operations