

Date: 06/04/2021

Black Board

Data Structures

Lecture # 8, 9&10

Topics:

Linked Lists I

{ contiguous vs linked structures
↓
Singly linked lists (SLLs)
SLL operations

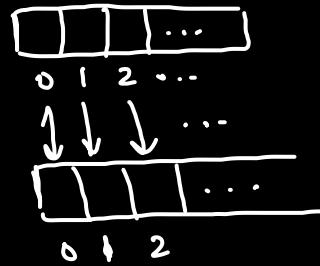
(1) Linked vs Contiguous structures

→ array

not contiguous

→ No direct access
(Con)

arr = 1000



arr[2] Direct access

↳ [1000 + h * 2] O(1)
(Pro)

} Copy of data
during resize O(n)
(Con)

→ Resize can be done quickly
(as fast as O(1)).

(2) Singly linked list



tail

700

Node

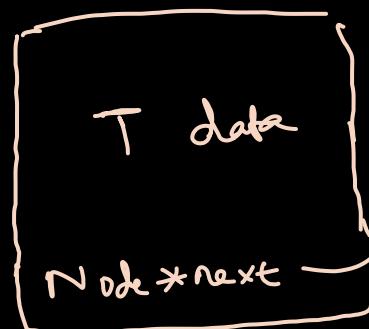
(Dynamically created
in the heap)

Structure of
a node

Empty list:



"List is made up of nodes"



pointer to
a node

(3). Linked List Operations

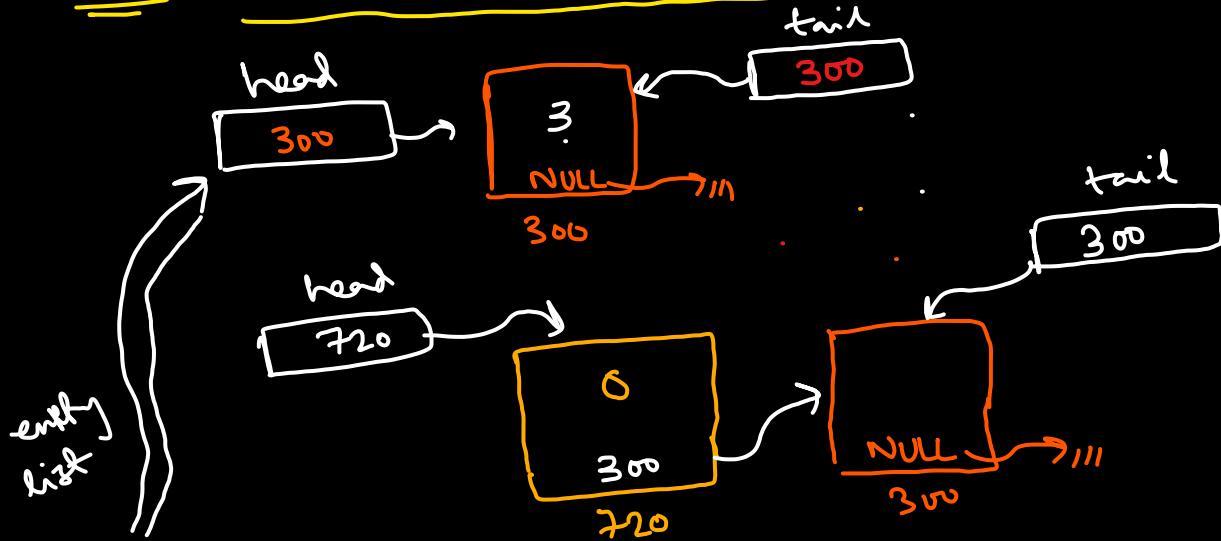
insert → insertAtHead
insert → insertAtTail

erase → eraseAtHead
erase → eraseAtTail ??!

insert → insertInOrder } O(n) worst case
erase → eraseKey

find → ??!

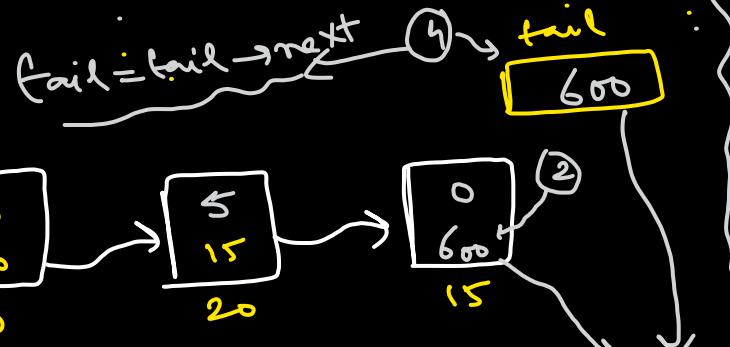
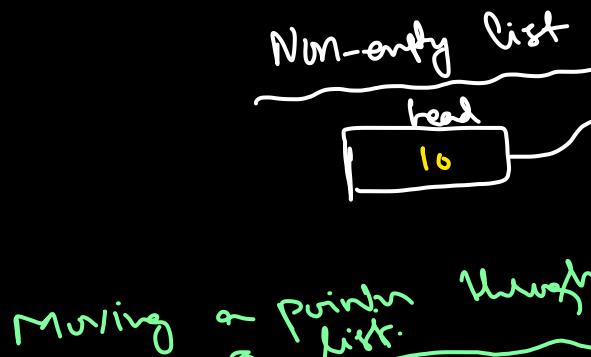
3.1 InsertAtHead, insertAtTail, insertInOrder



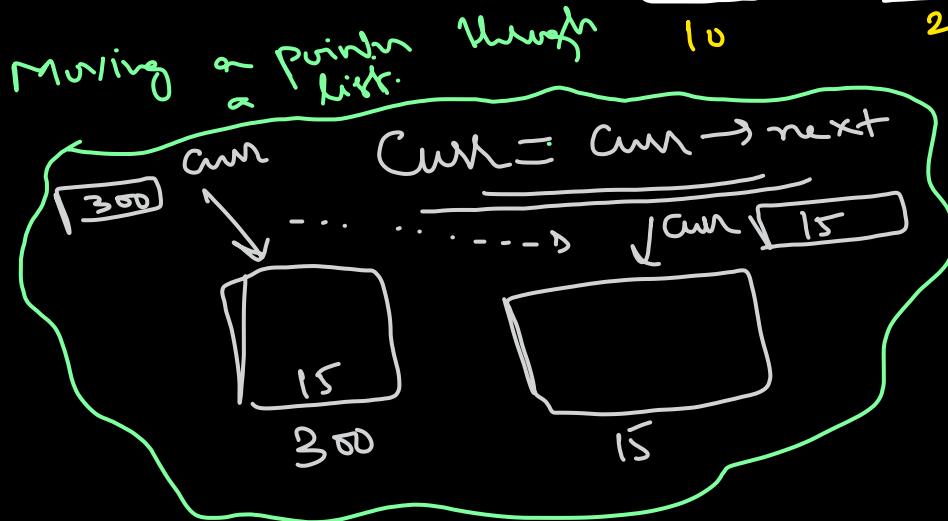
- l. insertAtHead(3)
l. insertAtHead(0)

- { (1) Create New node
(at 720) + add data
(2) → Set next of new
node to old head
(3) → Set head to the
new node address

InsertAtTail



- ① Create a new node + add data
- ② Set tail's next to the new node's address
- ③ Next of new node set to null.
- ④ move tail to the next node.



InsertInOrder

→ If the list is empty? It works in the same way as insertATHead & insertATTail.



→ l.insertInOrder(15) → Same actions as insertAtHead

→ l.insertInOrder(75) → Same actions as insertAtTail

→ l.insertInOrder(90) → Steps

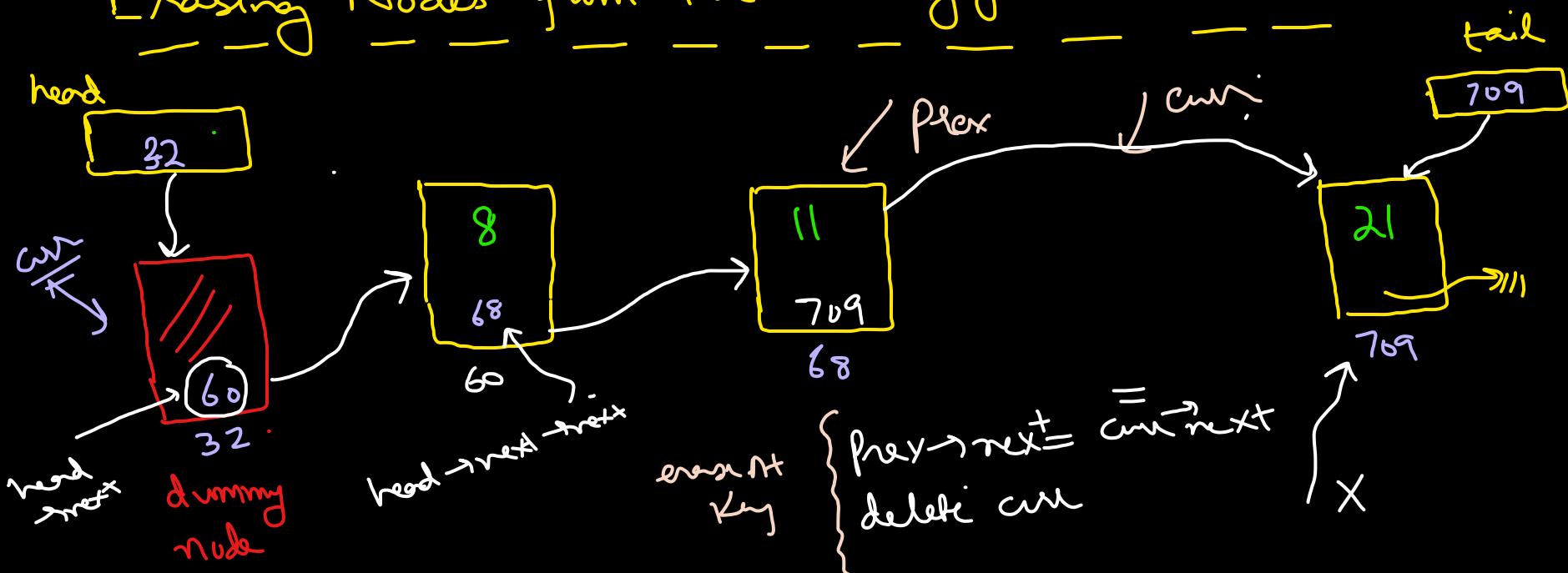
- {
- ① move curr to the node just after the new node (prev is one behind)
 - ② Create new node and use pointer temp to point to it-
 - ③ Set next of new node to curr.

④ put temp in the next of the prev node.

(using prev pointer)

Example of linked storage: Text Editor..
 Should I use an array or
 a linked list?

Erasing Nodes from the Singly Linked List



Erase at Head

Algo

Cases:

- Empty list → do nothing
- $\text{size} = 1$ → remove node → move tail to dummy.
- $\text{size} > 0$ → $\text{head} \rightarrow \text{next} = \text{head} \rightarrow \text{next} \rightarrow \text{next}$

Erase at Tail $O(n)$
Algo = {
 → First find previous of tail = ptail
 → Delete tail node
 → Set ptail → next = nullptr
 → Set tail to ptail.

Question: we wish to implement stack & queue using a
 singly link.

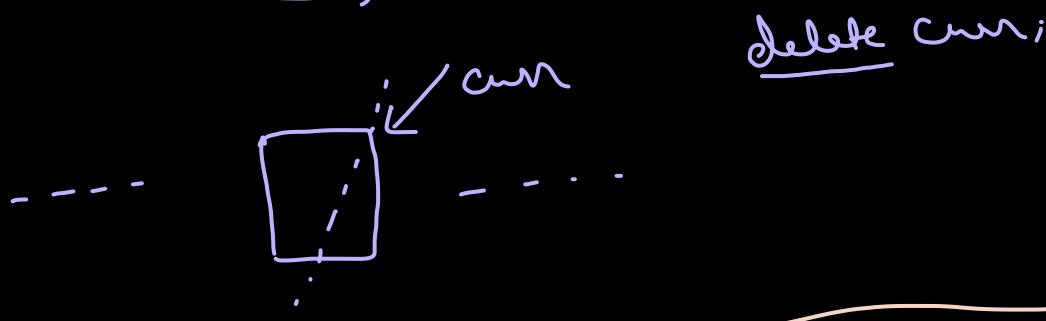
Stack: { Insert At Tail $\rightarrow O(1)$ (push) } not good
 LIFO { Erase At Tail $\rightarrow O(n)$ (pop) } choice

Insert At Head $\rightarrow O(1)$
 Erase At Head $\rightarrow O(1)$ } better choice.

Queue:
(FIFO)

$\left\{ \begin{array}{l} \text{insertAtTail}() \rightarrow O(1) \\ \text{eraseAtHead}() \rightarrow O(1) \end{array} \right\}$ good choice

Destructor of Singly linked list.



EraseAt Key

→ Find the node which contains the key
i.e. find pointer curr which points to it
(keep prev. one step behind curr).

