

Date: 06/04/2021

Black Board

Data Structures

Lecture # 8

Topics:

Linked Lists I

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

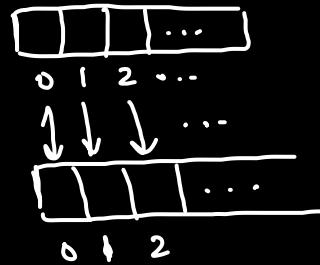
(1) Linked vs Contiguous structures

→ array

not contiguous

→ No direct access
(Con)

arr = 1000



arr[2]

Direct access
[1000 + h * 2] (Pro)

O(1)

} copy of data
during resize

O(n)
(Con)

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

(2) Singly linked list.

head

120

3

35

0

700

35

11

NULL

700

tail

700

Node

(Dynamically created
in the heap)

(we must always
maintain the
correct head
pointer.)

next pointer

Empty list:

head

0

→ ...

Structure of
a node



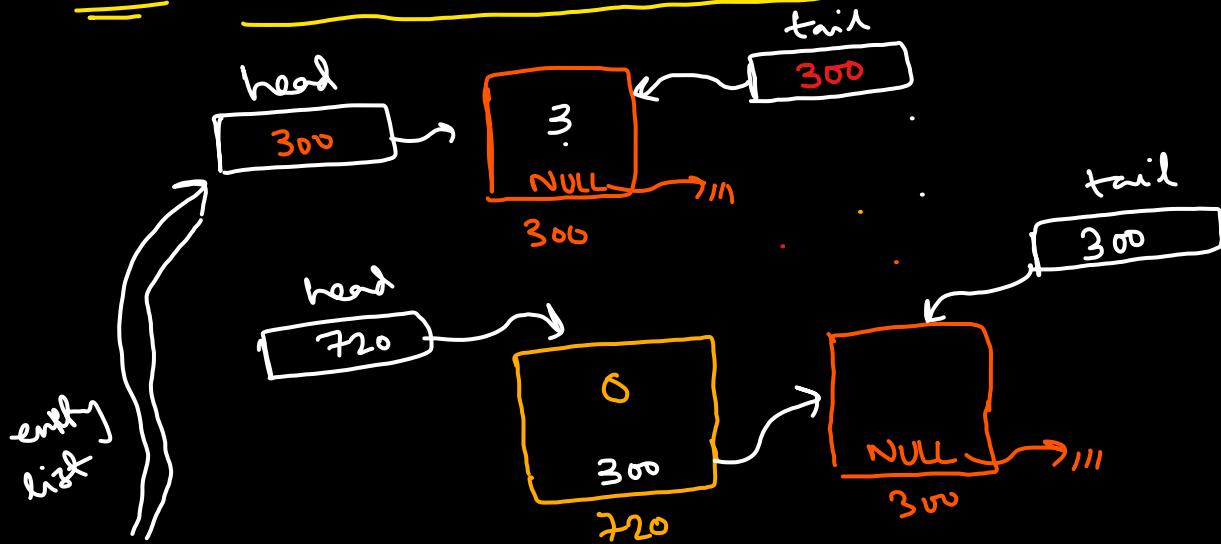
"List is made up of nodes"

(3). Linked List Operations

insert → insertAtHead
insert → insertAtTail
erase → eraseAtHead
erase → eraseAtTail
insert → insertInOrder
erase → eraseKey
find → ??!

insert → O(n) operations worst case
erase → ??!
insert → O(n) worst case
erase → O(n) worst case

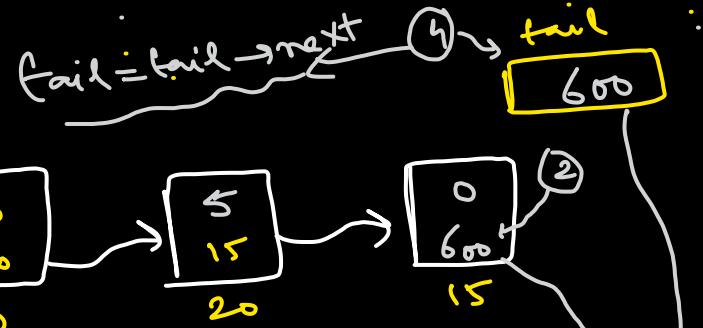
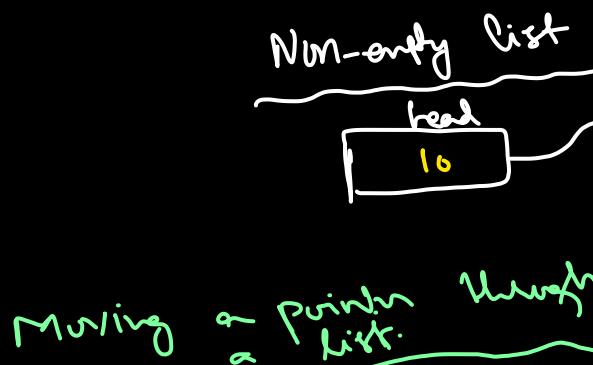
3.1 InsertAtHead, insertAtTail, insertInOrder



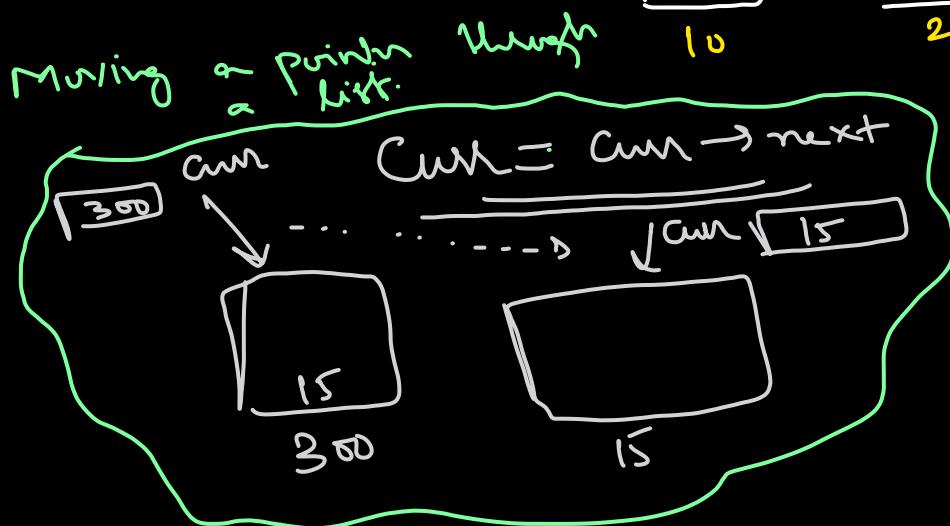
- l. insertAtHead(3)
l. insertAtHead(0)

- {
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

→ q.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?

