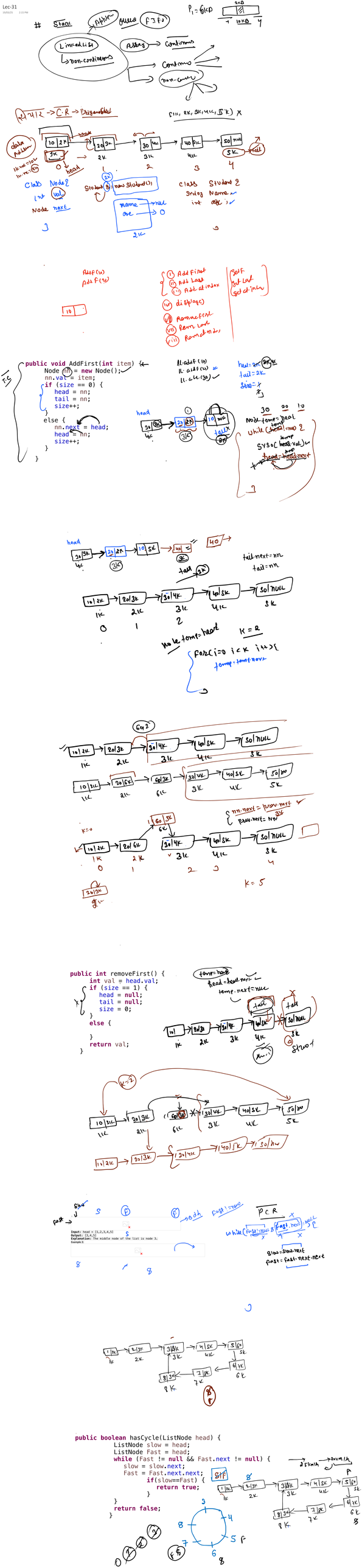
Lec-31
19/03/25 2:15 PM
Stack
Arrays
Queue
FIFO
Linked List
non-continuous
Array
Continuous
Continuous
non-Cont
Pi = 6KB
2KB
1KB
4
Ram, Rom, Hdd, Usb, SSD
Data Address
Node
Class Node2
int val
Node next
Class Student2
String Name
int age
Student2 s1 = new Student2()
name
one
null
0
AddF(10)
AddF(20)
(i) AddFirst
(ii) Add Last
(iii) Add at index
(iv) display()
(v) RemoveFirst
(vi) Remove Last
(vii) Remove at index
self
get Last
get at index
public void AddFirst(int item) {
Node nn = new Node();
nn.val = item;
if (size == 0) {
head = nn;
tail = nn;
size++;
}
else {
nn.next = head;
head = nn;
size++;
}
}
ll.addF(10)
ll.addF(20)
ll.addF(30)
head=3K
tail=2K
size=2
30 20 10
Node temp = head
While (temp!=null)
SysO(temp.val)
temp=temp.next
tail.next=nn
tail=nn
K=2
for(i=0 i<k i++){
temp=temp.next
}
nn.next=temp.next
temp.next=nn
K=0
K=5
public int removeFirst() {
int val = head.val;
if (size == 1) {
head = null;
tail = null;
size = 0;
}
else {
}
return val;
}
temp=head
head=head.next
temp.next=null
Input: head = [1,2,3,4,5]
Output: [3,4,5]
Explanation: The middle node of the list is node 3.
Example 2:
slow
fast
while(fast!=null && fast.next!=null)
slow=slow.next
fast=fast.next.next
public boolean hasCycle(ListNode head) {
ListNode slow = head;
ListNode Fast = head;
while (Fast != null && Fast.next != null) {
slow = slow.next;
Fast = Fast.next.next;
if(slow==Fast) {
return true;
}
}
return false;
}