DATA STRUCTURES AND ITS APPLICATIONS

Vandana M L

Department of Computer Science and Engineering

PES UNIVERSITY

DATA STRUCTURES AND ITS APPLICATIONS

Singly Linked List

Vandana M L

Department of Computer Science and Engineering

Deleting a node

There are 3 cases

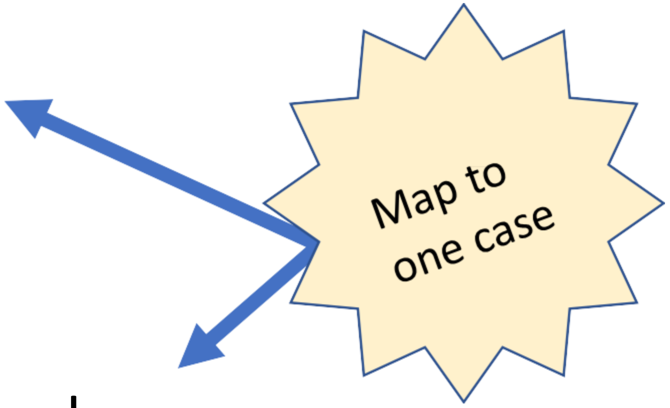
- Deleting first node
- Deleting last node
- Deleting a node at a given position

Deleting first node

Case 1: Linked list is empty

Case 2: Linked list with a single node

- delete the node
- set head to NULL



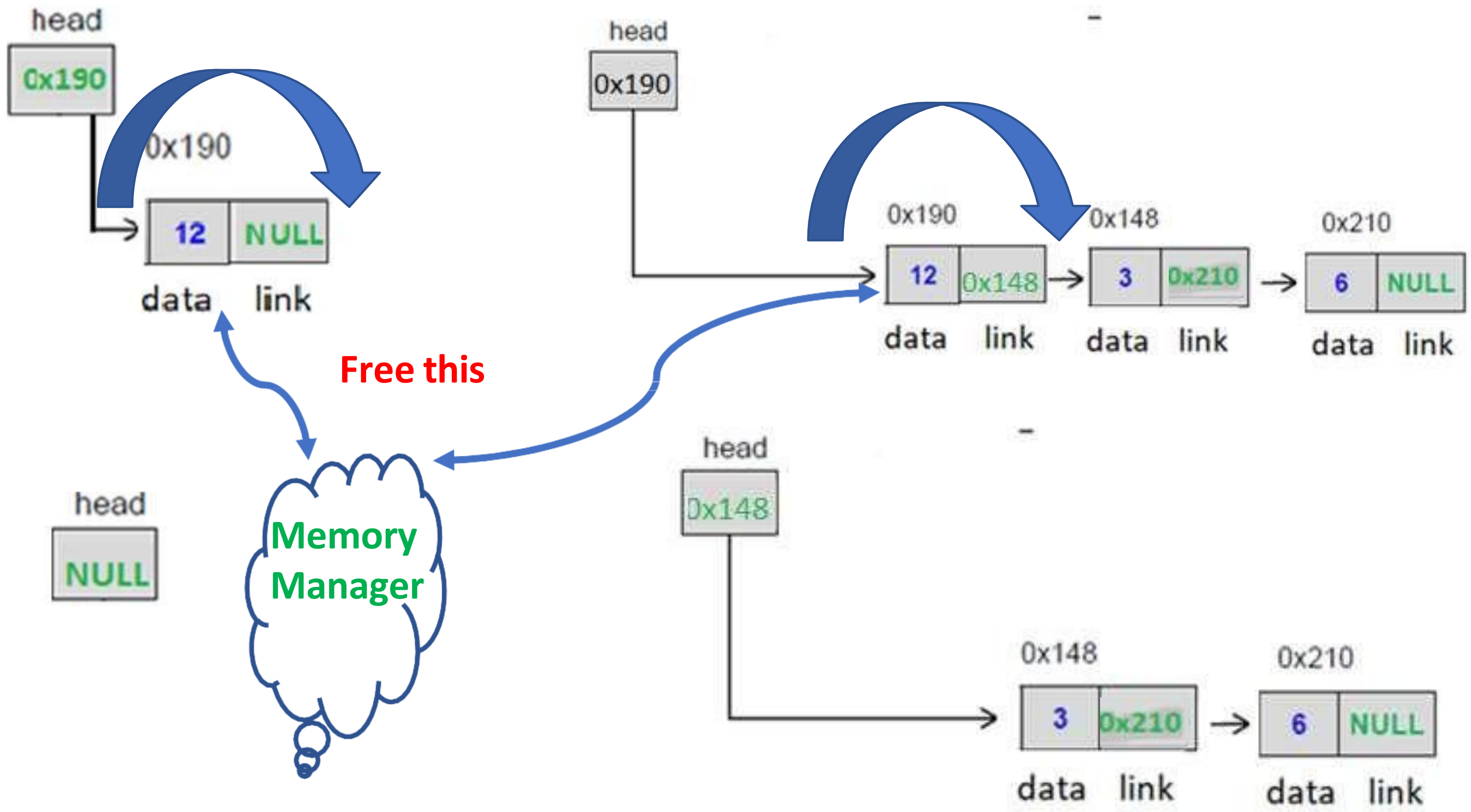
Map to one case

Case3:Linked list has more than one node

- Change head to point to second node
- Delete the first node

Deleting first node

Only one node in list **More than one node**



Deleting last node

Case 1: Linked list is empty

Case 2: Linked list with a single node

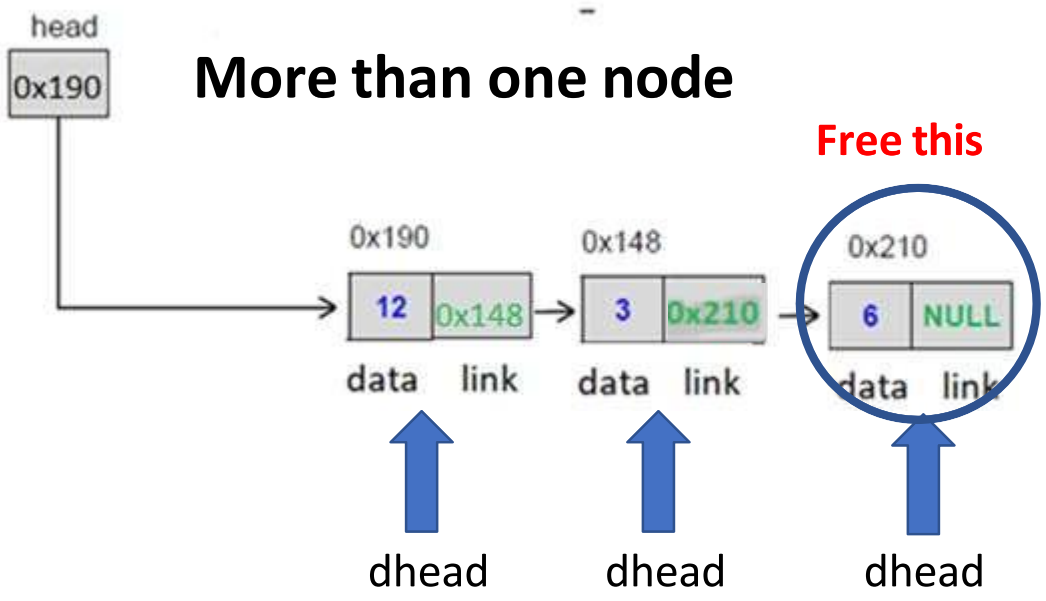
- delete the node
- set head to NULL

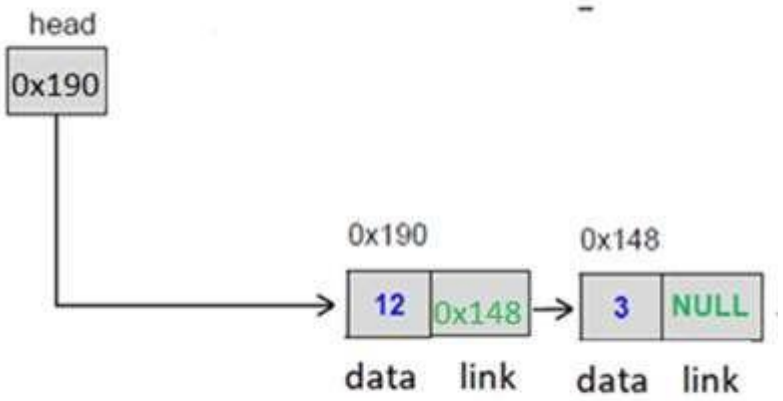
Case3:Linked list has more than one node

- Traverse the linked list to point to second last node
- Delete the last node
- Set link field of second last node to NULL

Deleting last node

More than one node



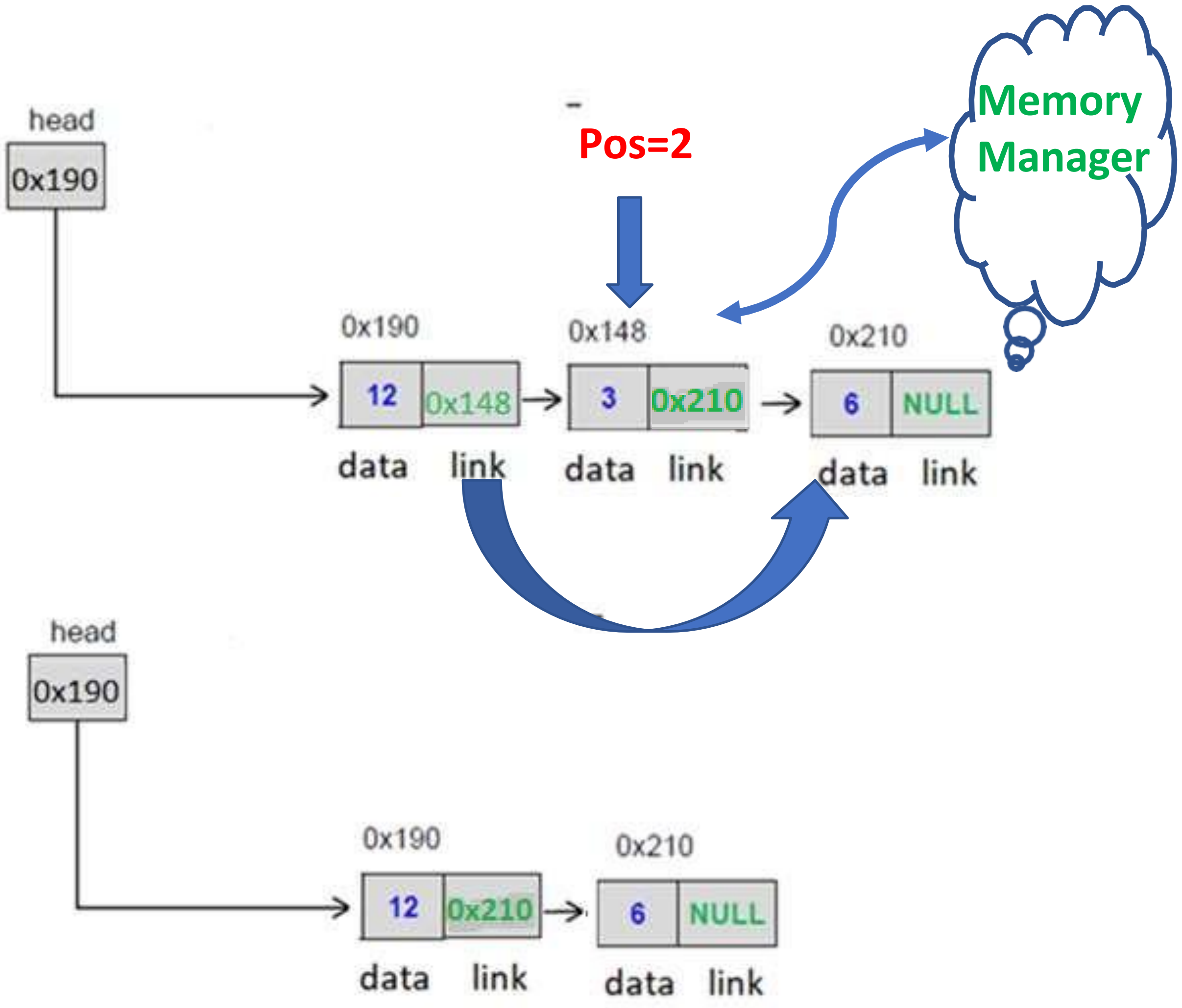


Deleting node from a given position

```
If the linked list is not empty
  If position is 1
  ➢     Delete from the front of the linked list
  Else
      If position is a valid position
          ➢ Traverse linked list to get the desired position
          ➢ keep track of previous node
          ➢ set previous node link field to link field of
            current node
          ➢ delete the current node
      Else
          ➢ print invalid position
```

Singly Linked List Operations

Deleting node from a given position



Singly Linked List delete operation

Apply the concepts to implement following operations for a singly linked list

- Delete a node with given key value
- Delete all alternate nodes
- Delete all the nodes (erase the linked list)

1. **Which is the correct sequence of operations to delete the head node of an SLL?**

a) head = head->next; free(head);

b) temp = head; head = head->next; free(temp);

c) free(head); head = head->next;

d) head->next = head; free(head);

1. **Which is the correct sequence of operations to delete the head node of an SLL?**

a) head = head->next; free(head);

b) temp = head; head = head->next; free(temp);

c) free(head); head = head->next;

d) head->next = head; free(head);

2. To delete the node after a given node p, which sequence is correct?

a) p->next = p->next->next; free(p);

b) temp = p->next; p->next = temp->next; free(temp);

c) temp = p; p = p->next; free(temp);

d) free(p->next); p->next = p->next->next;

2. To delete the node after a given node p, which sequence is correct?

a) p->next = p->next->next; free(p);

b) temp = p->next; p->next = temp->next; free(temp);

c) temp = p; p = p->next; free(temp);

d) free(p->next); p->next = p->next->next;

3. When deleting the last node in an SLL (without tail pointer), what is required?

a) Traverse to the second-last node, set its next to NULL, and free the last node.

b) Set head = NULL directly.

c) Free the last node and leave the second-last node unchanged.

d) No traversal is required.

3. When deleting the last node in an SLL (without tail pointer), what is required?

a) Traverse to the second-last node, set its next to NULL, and free the last node.

b) Set head = NULL directly.

c) Free the last node and leave the second-last node unchanged.

d) No traversal is required.

4. What happens if you attempt to delete from an empty SLL?

a) The program executes without issue.

b) A segmentation fault occurs due to NULL pointer access.

c) The head pointer is automatically initialized.

d) A new node is created.

4. What happens if you attempt to delete from an empty SLL?

a) The program executes without issue.

b) A segmentation fault occurs due to NULL pointer access.

c) The head pointer is automatically initialized.

d) A new node is created.

5. To delete a node at position n in an SLL (where n > 1), which of the following is correct?

a) Traverse n nodes and free the node at n.

b) Traverse n-1 nodes to find the previous node, then adjust its next and free the target node.

c) Swap data of the nth and (n-1)th nodes and free the (n-1)th node.

d) Always delete from the head if n > 1.

5. To delete a node at position n in an SLL (where n > 1), which of the following is correct?

a) Traverse n nodes and free the node at n.

b) Traverse n-1 nodes to find the previous node, then adjust its next and free the target node.

c) Swap data of the nth and (n-1)th nodes and free the (n-1)th node.

d) Always delete from the head if n > 1.

THANK YOU

Vandana M L

Department of Computer Science & Engineering

vandanamd@pes.edu