Computer Science & IT

Data Structure & programming

Linked List

Lecture No. 04

By- Abhishek Sir

Recap of Previous Lecture

- Topic
- Topic
- Topic
- Topic
- Topic

Single Linked list

Two pointer traversal

Slide

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

Double linked List.

Slide

Topic : Doubly Linked List

Head pointer

Slide

Topic : Doubly Linked List

Head pointer


```
typedef Struct dnode {
    int data;
    Struct dnode* next;
    Struct dnode* prev;
} Dnode;
```

Slide

Topic : Doubly Linked List

```
Dnode * getnode(int x){
    Dnode * temp = malloc(sizeof(Dnode));

    temp -> data = x;
    temp -> next = NULL;
    temp -> prev = NULL;
    return temp;
}
```

getnode return address of Node allocated by malloc()

Slide

Topic : Doubly Linked List

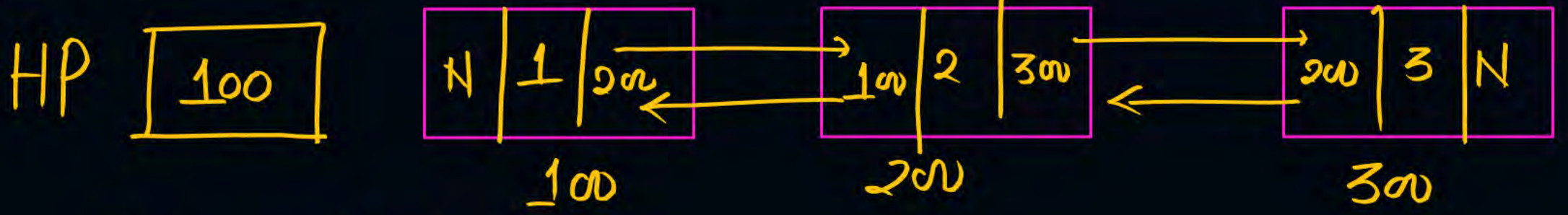
```
Dnode* Build123() {
    Dnode* temp1, temp2, temp3;
    temp1 = getnode(1);
    temp2 = getnode(2);
    temp3 = getnode(3)
    temp1→next = temp2;
    temp2→prev = temp1;
    temp2→next = temp3;
    temp3→prev = temp2;
    return temp1;
}
```

100 200 300

temp1 = 100 temp2 = 200 temp3 = 300

Slide

Topic : Doubly Linked List



HP is global variable

```
int main() {

    Dnode * HP;
    HP = Build123();

}
```

Slide

Topic : Doubly Linked List

Adding a Node at begin of Doubly Linked List

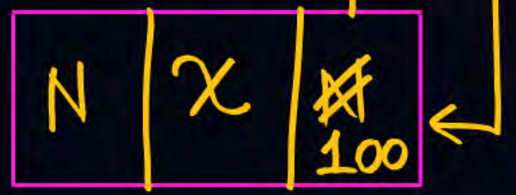
1. allocate a Node
2. if list is empty
   update HP
   return New Node
3. else
   update link
   update HP

Slide

Topic : Doubly Linked List

Head pointer

q = HP



P → next = q or p → next = HP

q → prev = P; // q → prev = NULL

Slide

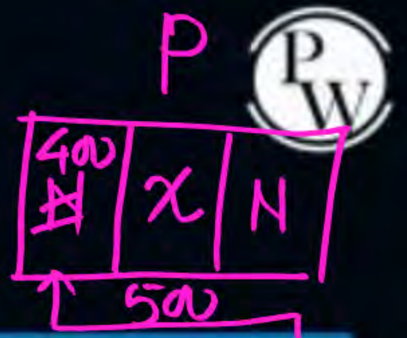
Topic : Doubly Linked List

```
void Addbegin(int x){
    Dnode *p=getnode(x);
→   Dnode* q=HP;
    if(HP==NULL){
        HP=P;
        return;
    }
    p→next = q;
    q→prev = p;
    HP=P;
    return;
}
```

Adding a Node at begining

Slide

Topic : Doubly Linked List



Adding a Node at the end of List.

q → next = P; ✓

P → prev = q;

1. Create a New Node P
2. check emptiness if empty return p update HP
3. travers till Last Node

q

Slide

Topic : Doubly Linked List

```
void Addend (int x){
    Node *p = getnode(x);
    Node*q = HP;
    if (q == NULL){
        HP = p;
        return;
    }
    while(q->next != NULL)
        q=q->next;
    q->next = p;
    p->prev = q;
    return
    }
```

Slide

Topic Question

The following C function takes a single-linked list of integers as a parameter and rearranges the element of the list. The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node {
        int value;
        struct node *next;
};
```

Topic Question

```
void rearrange (struct node *list){
strucut node *p, *q;
int temp;
if (!list || !(list-> next)) return;
p = list; q = list -> next;
while (q) {
temp = p ->value; p -> value = q -> value;
q -> value = temp; p = q ->next;
q = p ? p-> next : 0;
}
}
```

(A) 1, 2, 3, 4, 5, 6, 7
(B) 2, 1, 4, 3, 6, 5, 7
(C) 1, 3, 2, 5, 4, 7, 6
(D) 2, 3, 4, 5, 6, 7, 1

1 → 2 → 3 → 4 → 5 → 6 → 7

2 - 1 → 3 → 4 → 5 → 6 → 7

2 - 1 → 4 - 3 → 5 → 6 - 7

Topic Question

```
void rearrange (struct node *list){
strucut node *p, *q;
int temp;
if (!list || !(list-> next)) return;
p = list; q = list -> next;
while (q) {
temp = p ->value; p -> value = q -> value;
q -> value = temp; p = q ->next;
q = p? p-> next: 0;
}
}
```

(A) 1, 2, 3, 4, 5, 6, 7
(B) 2, 1, 4, 3, 6, 5, 7
(C) 1, 3, 2, 5, 4, 7, 6
(D) 2, 3, 4, 5, 6, 7, 1

1 → 2 → 3 → 4 → 5 → 6 → 7

2 1 3 4 5 6 7
P Q P Q P

2 1 4 3 5 6 7

Address q = p→next

Question

What is output of the program if the list contains 1→2 →3?

```
HP is global variable
void Dosomething(){
        node *q, *r, *s;
        q = HP;
        r = NULL;
                while( q!=NULL){
                        s = r;
                        r = q;
                        q = q→ link;
                        r→ link =s;
                }
                HP = r;
        }
```

(A) Same list

(B) List becomes 2→1→3

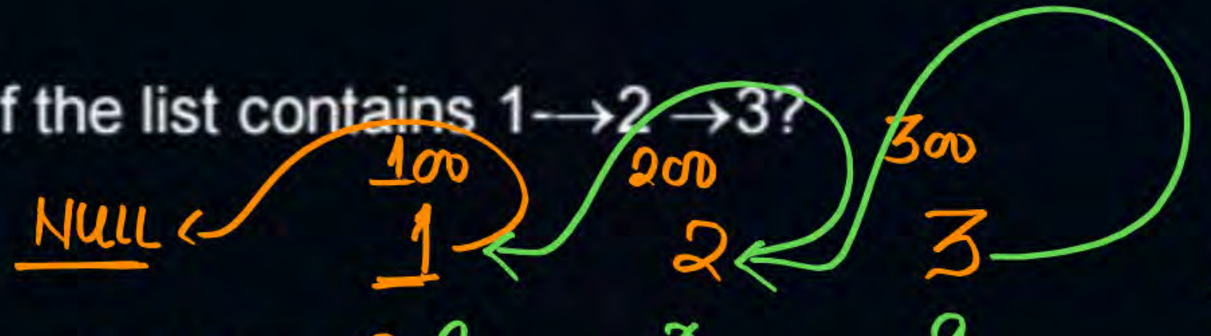
(C) List becomes 3→2→1

(D) Error

Question

What is output of the program if the list contains 1→2→3?

```
HP is global variable
void Dosomething(){
        node *q, *r, *s;
        q = HP;
        r = NULL;
            while( q!=NULL){
                    s = r;
                    r = q;
                    q = q→ link;
                    r→ link =s;
            }
            HP = r;
    }
```



r = NULL q s

① S = NULL
r = 100
q = 200

r s

② S = 100
r = 200
q = 300

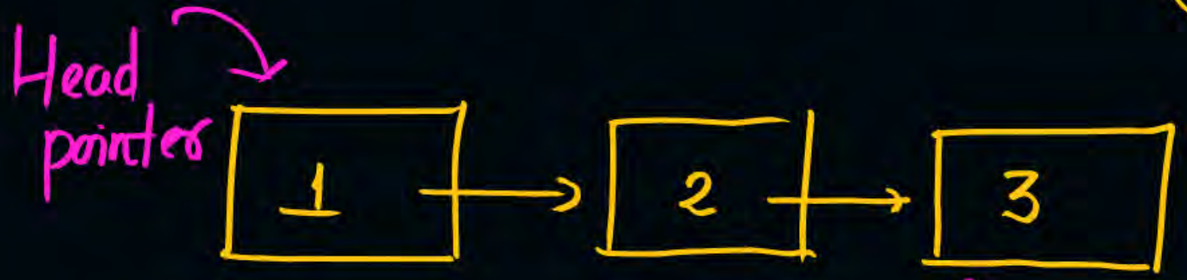
q r s

③ S = 200
r = 300
q = NULL

HP = 300

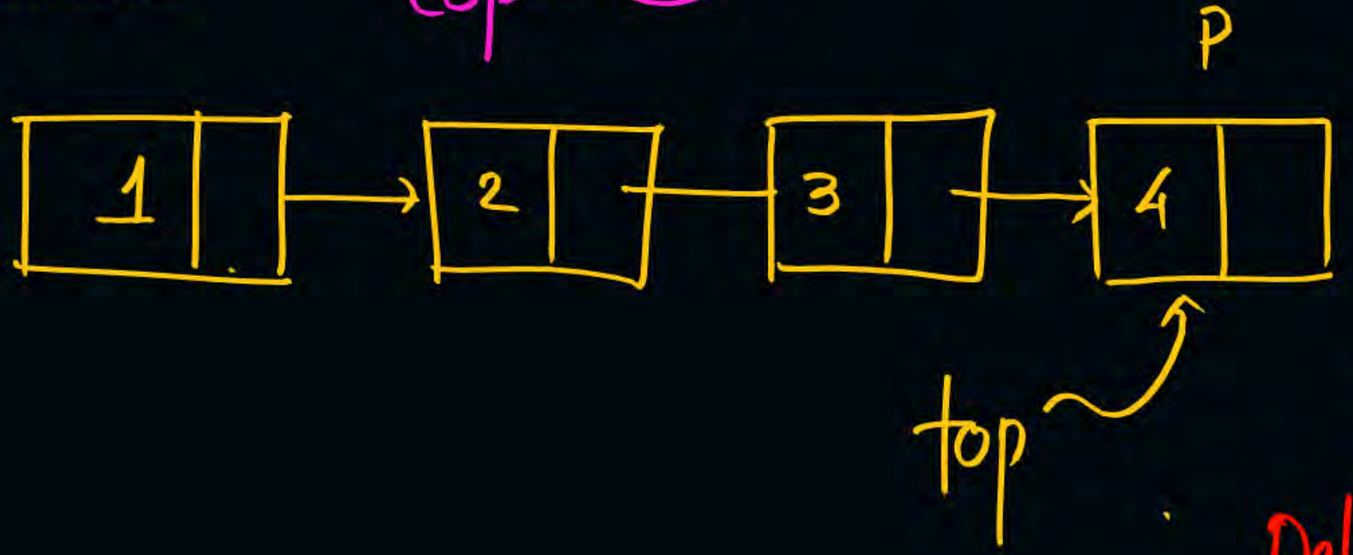
Reverse the Single Linked List

Q. if stack is maintaind using Single Linked list



(1) top points to last Node

Now Node Insertion



(1) push Depends upon the length of list (No)

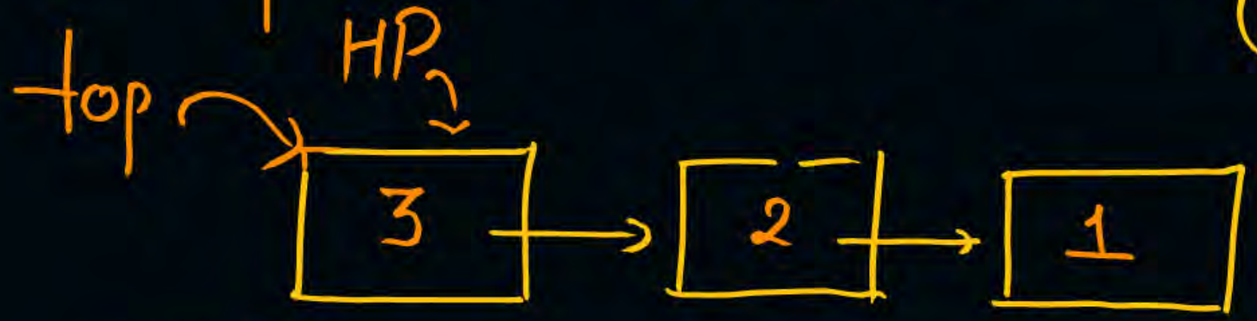
O(1)

(2) pop depends upon length of list

Delete the Last

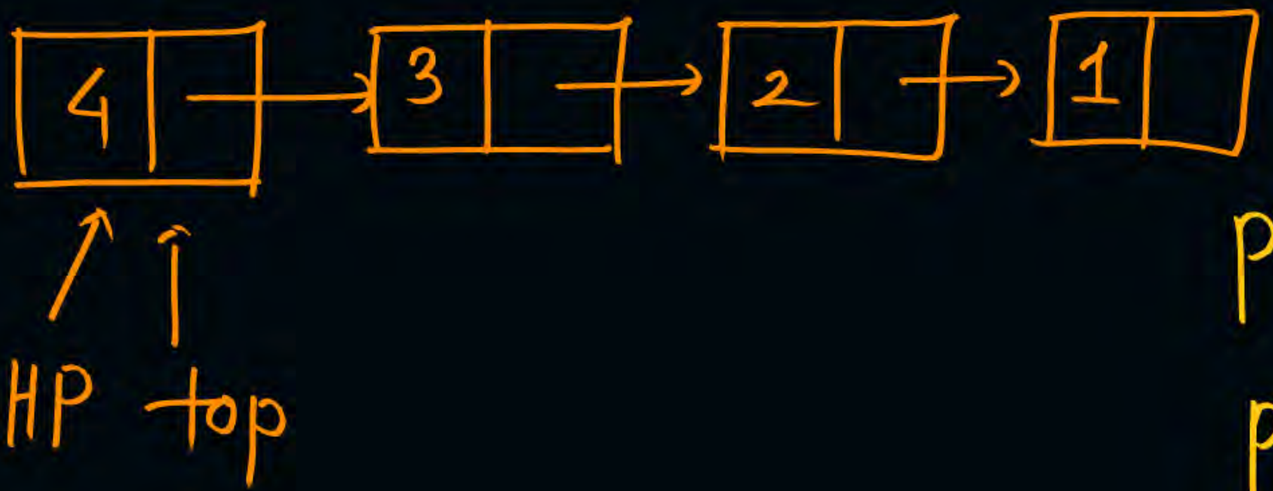
Algo O(n)

If stack is maintaind using Single Linked list



① Top points to first element

push(4)



② New Insertion at begining of list

push : No
pop : No

} None Depends upon length of list.

Topic : Single Linked List

HW

Let's understand the problem!

Given a singly linked list, write a program to find the middle node of the linked list. If the number of nodes is even, we need to return the second middle node.

Example 1: Input: 5->4->3->2->1, Output: 3

$5/2 = 3 \leftarrow \lceil 5/2 \rceil = 3$

Explanation: Here the number of nodes is 5, so there is one middle node which is 3.

Example 2: Input: 6->5->4->3->2->1, Output: 3

Explanation: The number of nodes is 6, where the first middle node is 4 and the second middle node is 3. So we need to return the pointer to the node 3.

4th element

Slide

2 mins Summary

Topic

Doubly linked List

Topic

get node, Add begin, Addend, Build 123

Topic

Reversal of singlelist

Topic

Stack using linked List

Topic

Slide

THANK - YOU

Slide