Linear Data Structure → Contd....

Linked List :→

It is a linear data structure which consists of an entity called Node.

Depending on type of node, there are three types of linked lists:→

(i) Singly linked lists
(ii) Doubly linked lists
(iii) Circular linked lists

| D | N | → | D | N | → null

Node; head; tail

Forward Traversal [head]

Insert
* head (front
* tail / end
* after specific

Delete
* head
* tail
* target

C++
```
class Node {
  Node * next;
  int data;
}
```

Java
```
class Node {
  Node next;
  int data;
}
```

Difference b/w Array & Linked List

Search O(1)
| 1 | 3 | 9 | 8 | 2 |
5th element
arr[4]

Insert n
| 1 | 3 | 4 | 5 | shift
2 O(n)
1 2 3 4 5

Search
head 1 → 2 → 3 → null
3rd node value
O(n)

Insert
0 → 1 → 2 → 3 null
3 steps
O(1)

curr
1 → 2 → 3 → null
head

a = 10   curr → data
curr = curr → next
int * ptr = &a;
*ptr;

0 — 1 → 2 — 3
0XX12
→ Node * head → address / mem
→ Node ** head → value

0 → 1 → 2 → 3 → null
head
newNode → 0
newNode → next = *head
*head = newNode;
**head value

last
0 → 0 → 0 → null
head   last   last
new Node → null
Node * last = * head;
while (last → next != nullptr) {
last = last → next;
}
last → next = new Node;
→ null

class
int data
Node * next
→ [address] *
0X101010
data → [value]

Node * head
0 — 0 — 0 → null
Node Node Node

Node *   a = [10] 1000
add
int a = 10
int * ptr = &a ; [1000]
point (ptr) = 1000
*ptr = 10
2000 &ptr = (2000)
int ** dptr = &ptr
2000

temp
1 → 2 → null
head   head
*head = (*head) → next
delete temp;

Delete End Node :→
(i) Empty → not possible
(ii) Single node →
if (head → next == null)
delete head
head = null;
x → null
h
→ null

slast   last
↓   1 → 2 → 3 → 4 → null
null   head   sl   l
Two Pointer Approach
→ while (last → next != nullptr) {
secondlast = last;
last = last → next;
}
delete last;
sl → next = null;

curr curr curr curr curr 6
1 → 2 → 3 → 4 → 5 → null
head

length = 0
count = 0 [1, 2, 3, 4, 5]
while (curr != nullptr)
count ++;
curr = curr → next;

5/2
= 2
middle
l = 5
l/2 = 2
ans = 2, <2
6/2 = 3
count = 0, 1, 2
while (count < ans)
temp = temp → next;
count ++

Feedback →
bigotic training.com

Session code : 10920