Name: Darsh Shah

Roll no: 43

Class: AI&ML

Single Linked list and doubly linked list

A. Introduction

A linked list is a data structure having node which stores an item and is connected linearly and are accessed using pointers. Unlike arrays the data is not stored contiguously.

B. Why use linked list over arrays?

Arrays can also be used to store linear data of similar types, but it has the following limitations.

1) Size: The size of an array is fixed. So, we must know the maximum number of elements in advance. Also, the memory used is equal to the upper limit of the array irrespective of the usage.

2) Inserting new element in an array is expensive because room must be created for new elements and to create room the existing elements needs to be shifted. Linked list if we have the head node then we can traverse to any node through it and insert new node at the required position. Deletion is also expensive with arrays until unless some special techniques are used.

C. Advantages of linked list

1) Dynamic size

2) Ease of insertion or deletion

D. Drawbacks of a linked list

1) Random access is not allowed: We have to access elements sequentially starting from the first node (head node). So, we cannot do binary search with linked lists efficiently with its default implementation.

2) Extra memory space for a pointer is required with each element of the list.

3) Not cache friendly. Since array elements are contiguous locations, there is locality of reference which is not there in case of linked lists.

E. Representation

A linked list is represented by a pointer to the first node of the linked list. The first node is called the head. If the linked list is empty, then the value of the head points to NULL.

Each node in a list consists of at least two parts:

1) data (we can store integer, strings, or any type of data).

2) Pointer (Or Reference) to the next node (connects one node to another)

In C, we can represent a node using structures.

Code:

Traversal of a link list:

// A simple C program for traversal of a linked list

#include <stdio.h>

#include <stdlib.h>

struct Node {

    int data;

    struct Node\* next;

};

void printList(struct Node\* n)

{

    while (n != NULL) {

        printf(" %d ", n->data);

        n = n->next;

    }

}

int main()

{

    struct Node\* head = NULL;

    struct Node\* second = NULL;

    struct Node\* third = NULL;

    head = (struct Node\*)malloc(sizeof(struct Node));

    second = (struct Node\*)malloc(sizeof(struct Node));

    third = (struct Node\*)malloc(sizeof(struct Node));

    head->data = 1;

    head->next = second;

    second->data = 2;

    second->next = third;

    third->data = 3;

    third->next = NULL;

    printList(head);

    return 0;

}

Output:

