NAME : Pratham Ashok Asrani

BRANCH/SECTION/ROLL_NO : CSE/3A/75

Link : https://github.com/PrathamAsrani/DSA_C/blob/master/Assignment_4.docx

Assume, Two Linked_list : L1 and L2 containing m and n number of elements respective.

Both list intersecting with each other and becoming a single list.

For e.g. :

L1 : 1,3,5,7,9,10,12,14

L2 : 2,4,6,8,10,12,14

In the above two list '10' is the intersecting point. After that the elements in both are same as they became single list.

To find the intersecting list we follow the below algo :

```
Intersecting_node(L1,L2){
    // Each value of List 1 will be compared with all the value of another list.
    // Nested for loop will used for comparison approach between the lists.
    For (int I = 0;I < m; I++){
        Struct Node *temp1= L1(head)
        For (int J=0; J<n; J++){
            Struct Node *temp2= L2(head)
            If(temp1 == temp2){
                Return temp1; }
            temp2 = temp2->link;}
        temp1 = temp1->link;}
Return NULL
} END.
```

Intersecting_node(L1,L2) // list 1 and list 2 given as Parameter or argument in the function

Performance Analysis:

Steps :

STEPS	FREQUENCY
1	m+1
2	1

3	mn + m
4	1
5	1
6	1
7	1

The standard time complexity:

Average Case : BigTheta of m*n (i.e. O(1)) // when the intersecting element is on the first place on both the list

Average Case : BigTheta of m*n (i.e. O(n2))

Worst Case : BigTheta of m*n (i.e. O(n2))