**Assignment No. – 5**

**Title:-** Link-list of positive and negative numbers.

**Problem statement:-** Write a C++ program to store set of negative and positive numbers using link-list. Write functions:

1. Insert numbers.
2. Delete nodes with negative numbers.
3. To create two more linked list using the list one containing all positive numbers, and other containing negative numbers.
4. For two lists that are sorted. Merge these two lists into third resultant list that is sorted.

**Learning Objectives:-** Implementation of merge sort to merge two sorted link lists.

**Pre-requisites:-** Operations of link list i.e. create, delete and transverse.

**Theory:-**

**\*Merge sort:-**

1) Merge sort is a sorting technique based divide and conquer technique.

2) With worse-case time complexity being 0(n log n), it is one of the most best algorithms.

3) Merge sort first divides the array into equal halves and then combines them in sorted manner.

**Algorithm:-**

Step-1) START.

Step-2) Store positive numbers in link list 1 and negative numbers in link list 2.

Step-3) Consider var a as array, n= total element in link list.

Step-4) Link list L1= as array= a [10]……….a[n/2].

Link list L2 as array= a [n/2+1]……….a[n].

L1= Merge sort (L1).

L2= Merge sort (L2).

return merge (L1, L2).

Step-5) Procedure for merge sort –

Var e as array which stores final result.

while (both arrays have elements)

begin

if (a [0] > b [0])

add b [0] to c

remove b [0] from b

else

add [0] to c

remove a [0] from a

end if.

end while.

While (b has elements)

Add b[0] to end of c.

Remove b [0] from b.

End while.

Return c.

Step-6) STOP.

Ex-1) 14 33 27 10 35 19 42 44.

>> Consider the above examples:

|  |
| --- |
| 14 33 27 10 35 19 42 44  List 1 List 2 |

Using merge sort we first divide the link list in two lists.

|  |
| --- |
| 14 33 10 27 35 19 42 44  List 1.1 List 1.2 List 2.1 List 2.2 |

Now, we divide the above list in two halves respectively.

Now, comparing elements of each list with each other.

27 > 10

10 27 Updated list.

|  |
| --- |
| 14 33 10 27 19 35 42 44  List 1.1 list 1.2 list 2.1 list 2.2 |

New lists are –

Now compare list 1.1 and 1.2 into 1 and 2.1 and 2.2 into 2.

New updated list after comparing each elements of 1.1 with 1.2 and 2.1 with 2.2 is-

|  |
| --- |
| 10 14 27 33 19 35 42 44  List 1 List 2 |

Now comparing these list and merge them into new list say Final List.

Final result after comparing the two list is:-

|  |
| --- |
| 10 14 19 27 35 42 44.    Final List. |

**\*Advantages of merge sort:-**

1) As its worst case complexity is 0 (n log n), it is very fast.

2) As compared to other sorting algorithms, its complexity is less.

**\*Disadvantages of merge sort:-**

1) Merge sort uses a lot of memory so it has high space complexity. As it divides and conquers, the memory management becomes tedious.

2) It does not go well when attempting to sort large data.

**\*Applications:-**

1) To calculate number of inversions.

2) E-commerce.

**\*Conclusion:-**Thus we, implemented merge sort to merge two link lists one containing positive numbers and other containing negative into one which was sorted.