

Lab Assignment: Merge K Sorted Lists using a Min-Heap

Objective

To implement an efficient algorithm to merge multiple sorted linked lists into a single sorted list using a Min-Heap data structure.

Problem Statement

Given K sorted linked lists, each containing N elements, the goal is to merge them into one single sorted linked list.

A Min-Heap is used to efficiently retrieve the smallest current element among all lists. By repeatedly extracting the minimum and inserting the next element from the same list, we can construct the merged sorted list.

Function Signature

```
struct Node* mergeKSortedLists(struct Node* lists[], int K);
```

Details

- Each of the K linked lists is already sorted in ascending order.
- Initialize a Min-Heap that can store one node from each list (based on their values).
- Insert the first node of each list into the Min-Heap.
- Repeat until the heap becomes empty:
 1. Extract the smallest node from the Min-Heap.
 2. Append this node to the result (merged) linked list.
 3. If the extracted node has a next node, insert that next node into the heap.
- The final merged list will be in ascending order.

Example Walkthrough

Input:

List 1: $1 \rightarrow 4 \rightarrow 5$

List 2: $1 \rightarrow 3 \rightarrow 4$

List 3: $2 \rightarrow 6$

Process:

1. Insert 1, 1, and 2 (heads of each list) into the Min-Heap.
2. Extract min (1), insert next from same list (4).
3. Extract next min (1), insert its next (3).
4. Extract next min (2), insert its next (6).
5. Continue until the heap is empty.

Output:

$1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 4 \rightarrow 5 \rightarrow 6$

Expected Output

The program should merge all K sorted linked lists into one sorted list using a Min-Heap and return the head pointer of the merged list.