

# Singly Linked List: Problem Set

Course: Introduction to Programming and Data Structures

**Dr. Laltu Sardar**

Institute for Advancing Intelligence (IAI),  
TCG Centres for Research and Education in Science and Technology (TCG Crest)

**tcg crest**

Inventing Harmonious Future

September 13, 2024

- 1 Singly Linked List
  - Some Linked List Problems

# Singly Linked List: Problem set

# Important Operations on Singly Linked Lists I

## ■ Insertion:

- **At the Beginning:** Inserting a new node at the start of the list.
- **At the End:** Inserting a new node at the end of the list.
- **At a Specific Position:** Inserting a new node after a given node.

## ■ Deletion:

- **From the Beginning:** Removing the first node of the list.
- **From the End:** Removing the last node (requires traversal to the last node).
- **From a Specific Position:** Removing a node located after a specific node.

## ■ Traversal:

- **Forward Traversal:** Accessing each node of the list from the head to the last node.

# Important Operations on Singly Linked Lists II

- **Search:**
  - **Search by Value:** Finding the first node containing a specific value.
  - **Search by Position:** Accessing the node at a particular index in the list.
- **Updating:**
  - Modifying the data stored in a specific node without altering the structure of the list.
- **List Reversal:**
  - Reversing the order of nodes in the list so that the first node becomes the last and vice versa.
- **Splitting:**
  - Dividing the list into two smaller lists at a given position.

# Important Operations on Singly Linked Lists III

- **Concatenation:**
  - Merging two singly linked lists into a single list.
- **Length Calculation:**
  - Counting the number of nodes present in the list.

# Some Linked List Problems

# Problem 1: Remove Duplicates from Sorted List

**Problem:** Given the head of a sorted linked list, remove all duplicates such that each element appears only once.

**Example:**

- Input:  $1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 3$
- Output:  $1 \rightarrow 2 \rightarrow 3$



## Problem 2: Critical Points in a Linked List

**Problem:** Identify critical points in a linked list. A critical point is defined as a node where the value is either strictly greater than both neighbors or strictly less.

**Example:**

- Input:  $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5$
- Output: Critical points at 3 and 2

# Problem 3: Cycle in a Linked List

**Problem:** Determine if a linked list has a cycle in it.

**Example:**

- Input:  $3 \rightarrow 2 \rightarrow 0 \rightarrow -4$  (cycle to node 2)
- Output: True (Cycle exists)

# Problem 4: Find Middle Element of Linked List

**Problem:** Given a singly linked list, return the middle node of the list.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$
- Output: 3

# Problem 5: Count Loop Length in Linked List

**Problem:** Find the length of the loop in a linked list if it exists.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2$  (cycle to node 2)
- Output: 3 (Length of loop is 3)

# Problem 6: Sort a linked list

**Problem:** Sort a linked list using any sorting algorithm (E.g., merge sort, quick sort, etc.).

**Example:**

- Input:  $4 \rightarrow 2 \rightarrow 1 \rightarrow 3$
- Output:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

# Problem 7: Separate Even and Odd values in a linked list

**Problem:** Separate the even and odd values of the linked list, maintaining their relative order.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$
- Output:  $2 \rightarrow 4 \rightarrow 1 \rightarrow 3$

# Problem 8: Find Intersection of two linked lists

**Problem:** Find the node where two singly linked lists intersect.

**Example:**

- List 1:  $1 \rightarrow 9 \rightarrow 1 \rightarrow 2 \rightarrow 4$
- List 2:  $3 \rightarrow 2 \rightarrow 4$
- Output: Node with value 2

# Problem 9: Rotate the List

**Problem:** Rotate the linked list to the right by  $k$  places.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ,  $k = 2$
- Output:  $4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$



# Problem 10: Reverse The Linked List

**Problem:** Reverse the entire linked list.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$
- Output:  $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$

# Problem 11: Reverse the Segment

**Problem:** Reverse a portion of the linked list from position  $m$  to  $n$ .

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ,  $m = 2$ ,  $n = 4$
- Output:  $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$

# Problem 12: Find Next Smaller value in Linked List

**Problem:** For each node, find the next node in the linked list with a smaller value.

**Example:**

- Input:  $5 \rightarrow 3 \rightarrow 8 \rightarrow 2$
- Output:  $3 \rightarrow 2 \rightarrow 2 \rightarrow -1$

# Problem 13: Flatten a Linked List

**Problem:** Flatten a linked list where each node contains a pointer to another linked list.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3$ , and 1 points to list:  $4 \rightarrow 5$
- Output:  $1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 3$

# Problem 14: Reverse m size groups

**Problem:** Reverse the nodes in a linked list in groups of size m.

**Example:**

- Input:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ,  $m = 3$
- Output:  $3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 5$



**THANK YOU**

**FOR YOUR ATTENTION**

**tcg crest**

Inventing Harmonious Future

Dr. Laltu Sardar

[laltu.sardar@tcgcrest.org](mailto:laltu.sardar@tcgcrest.org)

<https://laltu-sardar.github.io>