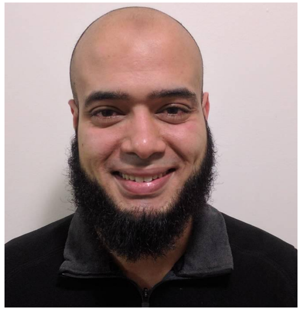
Data Structures

SLL Homework 4

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Problem #1: Arrange odd & even nodes

- This problem is not about nodes values, but their positions (odd & even)
- Rearrange the nodes so that, odd nodes comes first and even nodes comes last
- E.g. if list is 10, 20, 3, 7, 15: Nodes (10, 3, 15) are at odd positions
- 1, 2, 3, 4 ⇒ 1, 3, 2, 4
- 1, 2, 3 ⇒ 1, 3, 2
- 1, 2, 3, 4, 5, 6, 7 ⇒ 1 3 5 7 2 4 6
- 11 33 55 4 50 17 8 ⇒ 11 55 50 8 33 4 17

Problem #2: Insert alternating ?

- Implement void insert_alternate(LinkedList &another)
- The function insert the values from another in an alternating way with self
- E.g. if list1 = 1, 2, 3 and list2 = 4,5,6⇒ 1 4 2 5 3 6
- {1, 2, 3}, {4} ⇒ {1, 4, 2, 3}
- {1, 2, 3} {4, 5, 6, 7, 8}⇒ 1 4 2 5 3 6, 7, 8
- {}, {1, 2, 3} ⇒ {1, 2, 3}

Problem #3: Adding 2 HUGE numbers

X

- Assume we want to represent number 157 as linked list
  - It is helpful to have list as 7 -> 5 -> 1
  - This makes it easy to build and use in math operations
- Implement method: void add_num(LinkedList &another)
- It adds another number to its **current** values
- Let’s say current list is {1, 2, 3} representing 321
- Another is: {4, 5, 3} representing 354
- After the addition the list became: 5 7 6   {represents 675}
- {9, 6, 5} + {8, 7, 6, 4, 5, 7, 8, 9} ⇒ {7, 4, 2, 5, 5, 7, 8, 9}
- Notice: numbers are huge. Don’t convert to integer

Problem #4: Remove all repeated

- Given linked list of **sorted** integers, keep only nodes that **never repeated** and remove everything else (duplicate nodes)
- Input: 1, 1, 2, 2, 2, 3, 5 ⇒ {3, 5}   both 1 and 2 are repeated
- Input: 1, 1 ⇒ {}
- Input: 1, 1, 2, 2, 2 ⇒ {}
- Input: 1, 1, 2, 2, 2, 5 ⇒ {5}
- Input: 1, 2, 2, 2, 3 ⇒ {1, 3}
- Caution: Coding this problem may drain your time
  - Think about several test cases
  - Draw & verify!

Problem #5: Reverse Chains

- Implement: void reverse_chains(int k)
- Instead of reversing the whole list, you reverse only each consecutive k nodes
- {1,2,3,4,5,6}, k = 6 ⇒ 6 5 4 3 2 1 [normal reverse]
- {1,2,3,4,5,6}, k = 3 ⇒ 3 2 1 **6 5 4**
- {1,2,3,4,5,6, 7}, k = 2 ⇒ 2 1 4 3 6 5 7

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."