Data Structures

DLL Homework 2

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

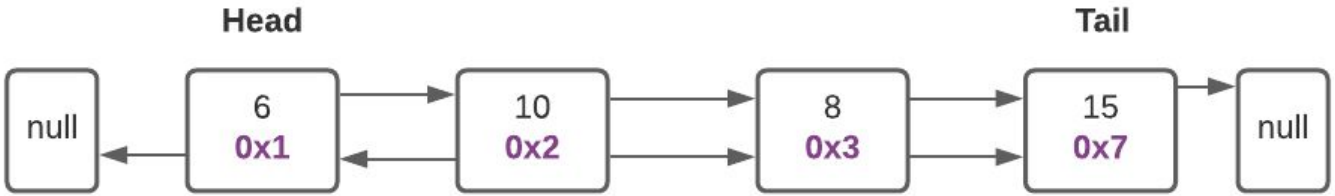
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: Find the middle

- Given a linked list, we would like to find its middle value
  - In an odd length list, e.g. {1, 2, 3, 4, 5}, the middle value is 3
  - In an even length list, e.g. {1, 2, 3, 4, 5, 6}, the middle values are {3, 4}.
    - We need the second one, so answer is {4}
  - Return None for empty list!
- Provide **2 implementations**, but consider:
  - You can't iterate on the list more than once!
  - Don't use the length variable!
- First: Use your doubly linked list
- Second: Solve it only with the next node (SLL). Don't use the previous
  - ~5 lines of code.

Problem #2: Swap forward with backward

- Given K, find the kth node from from the front and back of the list
  - K is in range [1, total nodes
  - Swap them (**address** not values).
  - For example: for k = 1, we swap head (0x1) and tail (0x7)
  - For example: for k = 2, we swap nodes 0x2 and 0x3 ⇒ **(6/0x1), (8/0x3), (10/0x2), (15/0x7)**
  - Trick cases. Think and consider
- def swap_kth(self, k)



Problem #3: Reverse list nodes

- Given a list, reverse all its nodes (addresses)
- E.g. {1, 2, 3, 4, 5} ⇒ {5, 4, 3, 2, 1}
- def reverse(self)

Problem #4: Merge lists

- Assume we have 2 sorted linked lists, of sizes n and m
- We would like to merge them together in O(n+m) but remain sorted
- **def** merge_2sorted_list(self, other):
  - We don’t care about list other after the function call
- E.g. list1 {10,20,30,40,50} and list2 {15,17,22,24,35}
  - ⇒ 10 15 17 20 22 24 30 35 40 50
- Consider the different cases!

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."