

Problem Domain:

- Write a method `.insertAfter(value, newVal)` that takes in a search value and a new value and inserts the new value after the search value in a linked list.

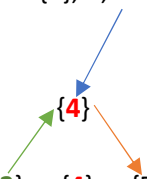
Input:

1. Value
2. newVal

Output:

1. Void

Visual:

- Input {1} -> {2} -> {3} -> {5}, 3, 4
- 
- Output {1} -> {2} -> {3} -> {4} -> {5}

Edge Case:

1. Value Type
2. Value Null
3. Value not found
4. Empty List

Big O:

1. Time: $O(n)$
2. Space: $O(1)$

Algorithm:

1. Create a method that takes a search value and new value as parameters
2. Declare a new node and set it equal to head of list
3. Declare a new node and set the value equal to passed in new value
4. Iterate over the list
5. Check to make sure the list isn't empty and if it is set the head of the list equal to the new Node
6. When current node value is equal to search value
 - Set new node next pointer equal to the next node in the list
 - Set the current node's pointer equal to new node

Pseudo Code:

Algorithm insertAfter(value, newVal)

- node current = list.head
- node newNode = new Node(newVal)
-
- If current != null
 - While current != null
 - If current.value = value
 - newNode.Next = current.next
 - current.next = newNode
 - Break;
 - current = current.next
- else
 - current = newNode

Verification:

- Input {1} -> {2} -> {3} -> {5}, **3, 4**

Current	Current.Next	NewNode.Next	Boolean	List
1	2	null	1 == 3 false	{1} -> {2} -> {3} -> {5}
2	3	null	2 == 3 false	{1} -> {2} -> {3} -> {5}
3	5	null	3 == 3 true	{1} -> {2} -> {3} -> {5}
3	= 4	= 5	break	{1} -> {2} -> {3} -> {4} -> {5}