



DATA STRUCTURES AND ALGORITHMS
ENGR-UH 3510

PRE LAB 4

Basic Data Structures

Single Linked List

1 Theory

The following section describe one of the operations that can be applied to a singly-linked list. In addition to what we already did in the previous lab with push(enqueue) and pop(dequeue).

Insertion (put()) Operation

Adding a new node in the linked list involves more than one step. First, create a node using the same structure. With the enqueue() operation this new node is added at the right end of the list. In case of inserting a new node, the location where it should be inserted is specified; the next series of figures shows the exact steps that should be followed. This procedure is slightly more elaborated than enqueueing a node at the end of the list.

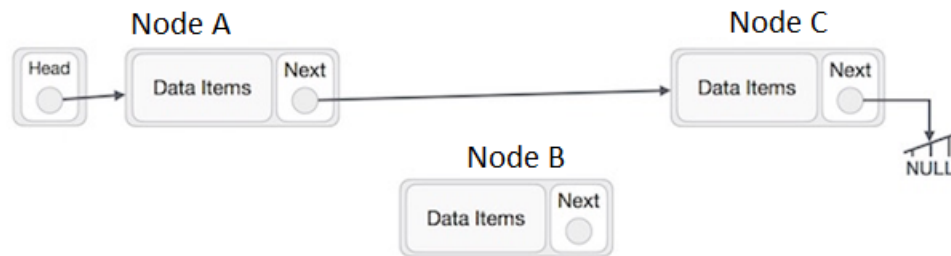


Figure 1: Insert Step #1

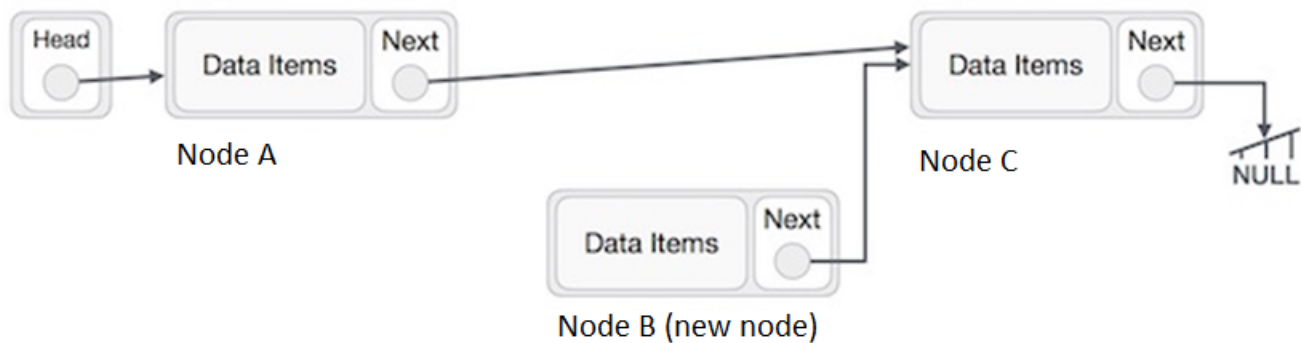


Figure 2: Insert Step #2

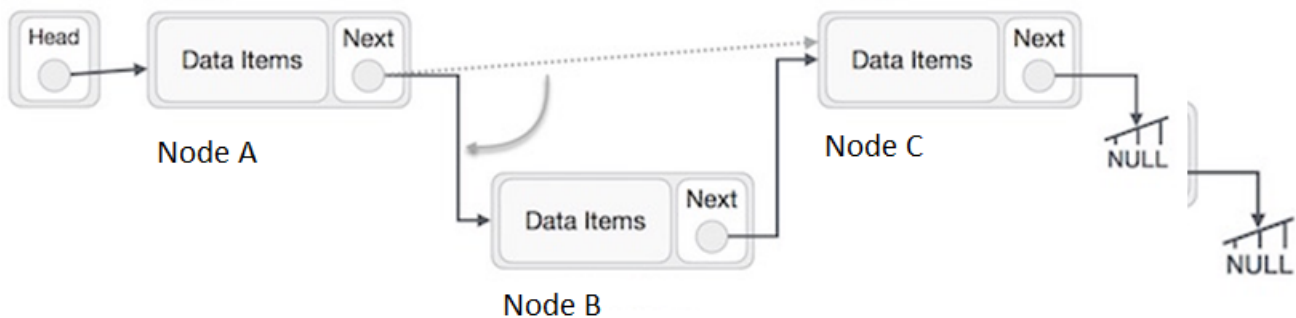


Figure 3: Insert Step #3

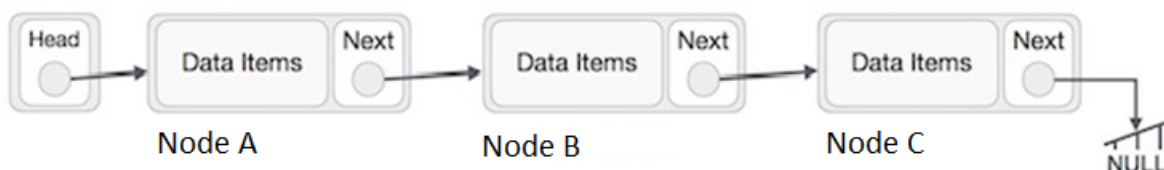


Figure 4: Insert Step #4

Similar steps should be taken if the node is being inserted at the beginning of the list. While inserting it at the end, the second last node of the list should point to the new node and the new node will point to NULL.

2 Assignment

Construct a singly-linked list to store integer values.

1. Define the structure that is needed following the below definition, if you assume that you need additional attributes you may add them.

```
Node {  
    int data;  
    Node next;  
}
```

Listing 1: Node class

```
List {  
    Node head;  
}
```

Listing 2: Queue class

2. Implement a put function that enables insertion of new items to the list in a selected position.
3. Implement a print functions that displays the contents of the stack.
4. Implement a main functions that demonstrates the aforementioned functionality. It is up to you to define the sequence of operations that are needed to be done in order to best demonstrate your code. Always use the needed print outs.