Data Structures

Linked-list-based Stack

Mostafa S. Ibrahim
Teaching, Training and Coaching for 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)

Using Linked list

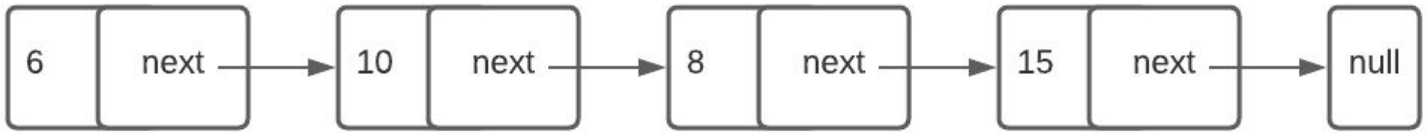
- Similar to the array implementation, we can push in the stack elements: 15, 8, 10, 6
- Which one is enough? SLL or DLL?
- How can we implement it?

Building Stack

- We can use a **single node for the head**, which creates reversed elements
  - Here we added in the order:
  - 15, 8, 10, 6
  - Review homework
- The nice thing, this fits with stack, which reverse elements!

```
class Stack:
    def __init__(self):
        self.head = None
        self.length = 0

    def push(self, value):
        # By design: always new node = head
        # Great match with stack!
        item = Node(value)
        item.next = self.head
        self.head = item
        self.length += 1
```



Remaining Functions

```
def pop(self):
    assert self.head, 'No items!'

    node = self.head
    self.head = self.head.next
    self.length -= 1

    return node.data

def peek(self):
    assert self.head, 'No items!'
    return self.head.data

def isEmpty(self):
    return self.length == 0

def size(self):
    return self.length
```

Array vs Linked-list for stack

- Compare the time/memory order of both data structures
- What are the advantages of each type?
- Compare and contrast the efficiency of different functions (and possible additional functionalities) for both the array and linked list based stack classes

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."