Data Structures

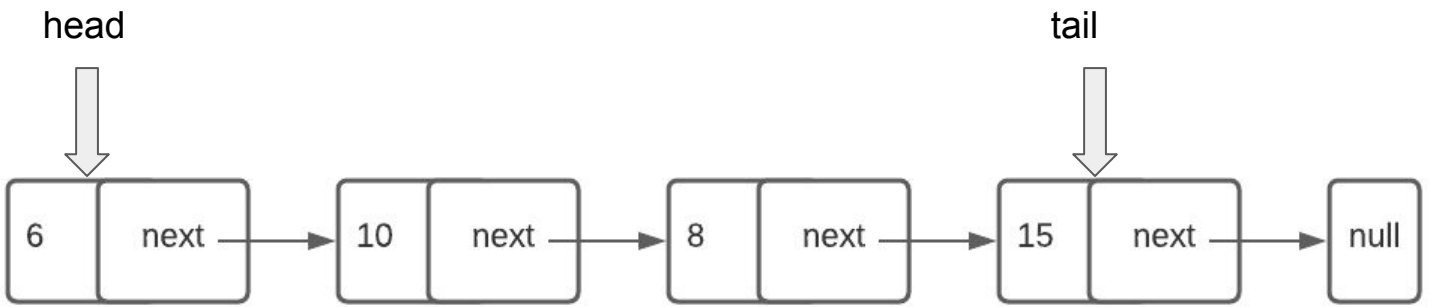
Linked-list-based Stack

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)

Using Linked list

- Similar to array, we can push in stack elements 15, 8, 10, 6
  - Using Insert front
  - But now, NO LIMIT to the number of nodes (except machine limit)
- We can use either SLL or DLL
- Can u notice why using head only is enough and intuitive?

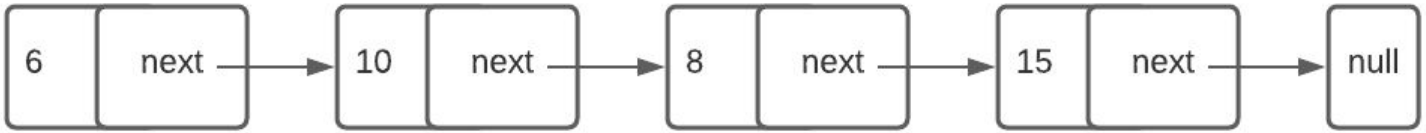


Recall SLL Homework #1

- Using a head only, we can build SLL
- However, nodes have to be **reversed**
- Here we added in order:
  - 15, 8, 10, 6

```
class LinkedList {
private:
    Node *head { };

public:
    void add_element(int value) {
        Node* item = new Node(value);
        item->next = head;
        head = item;
    }
```



SLL Stack

- Let’s create a stack based on linked-list style: with only single head

```
6  class Stack {
7  private:
8      // Tip: We can make this struct internally
9      struct Node {
0          int data { };
1          Node* next { };
2          Node(int data) :
3                  data(data) {
4          }
5      };
6
7      Node *head { };     // Points to Top
8
```

Push function

- The previously add element!

```
void push(int value) {
    // By design: always new node = head
    // Great match with stack!
    Node* item = new Node(value);
    item->next = head;
    head = item;

    // Tip: This code works as long as machine has more RAM
    // In industry: You may check if return is null (can't create) or not.
}
```

Remaining

- If you did well with linked-lists, all that should be direct!

```
int pop() {
    assert(!isEmpty());
    int element = head->data;
    Node* temp = head;
    head = head->next;
    delete temp;
    return element;
}

int peek() {
    assert(!isEmpty());
    int element = head->data;
    return element;
}

int isEmpty() {
    return !head;
}
```

Array vs Linked-list

- Compare the time/memory order of both data structures?
- What are the advantages of each type?
- What kind of functionalities extension could be more efficient for one of them?

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."