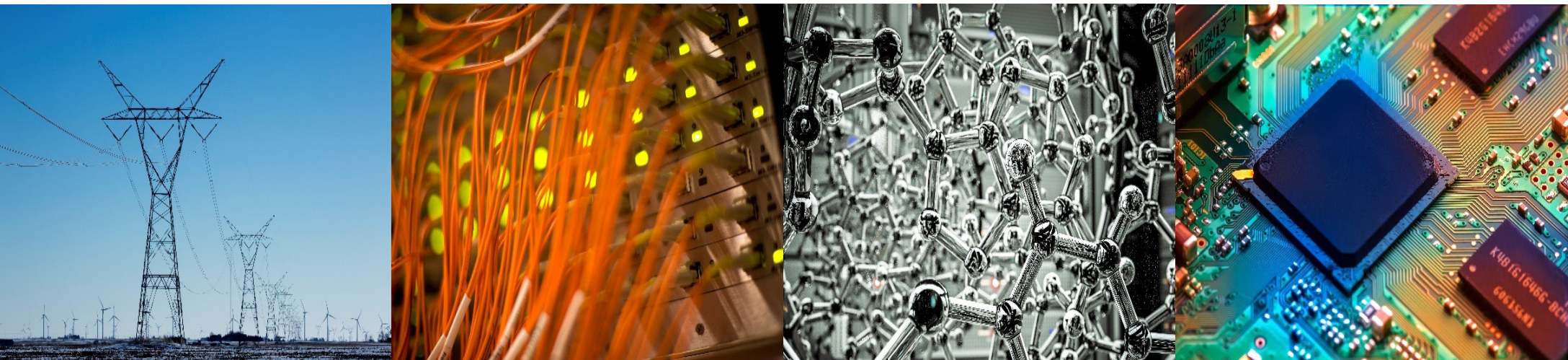


ECE 220 Computer Systems & Programming

Lecture 18 – Problem Solving with Linked Lists

March 31, 2026



- MT2 is this Thursday, lecture will be cancelled
- Check course website for your MT2 room assignment

I ILLINOIS

Electrical & Computer Engineering

GRAINGER COLLEGE OF ENGINEERING

When do we need to use a double pointer?

```
int main() {
    /* add new node to a (sorted) linked list in main */
    Node *head;
    head = (Node *)malloc(sizeof(Node));
    head->UIN = 12345;
    head->GPA = 4.0;
    head->next = NULL;

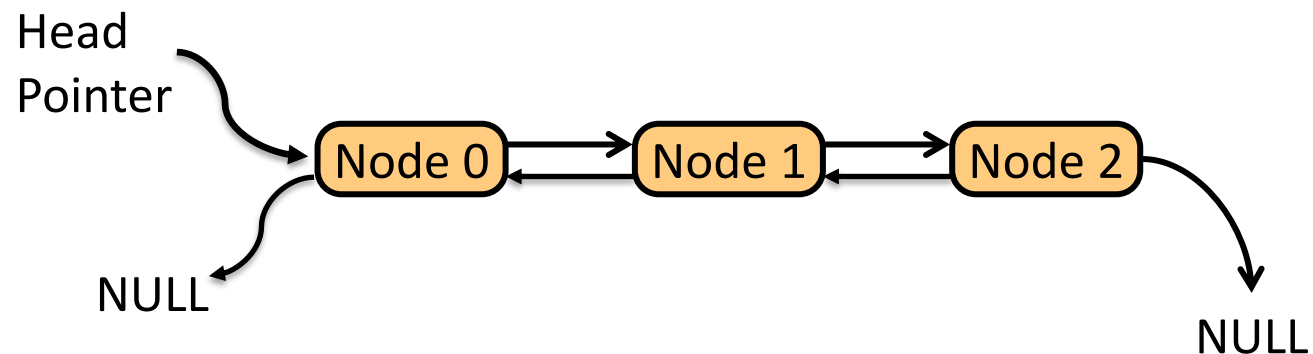
    /* add new node by calling another function */
    add_node(&head, 11111, 3.0);

    /* memory deallocation omitted here for simplicity */
    return 0;
}
```

x6001	
x6002	
x6003	
x6004	
x6005	
x6006	
x6007	
x6008	
x6009	head
	Main's Bookkeeping Info

Doubly linked list

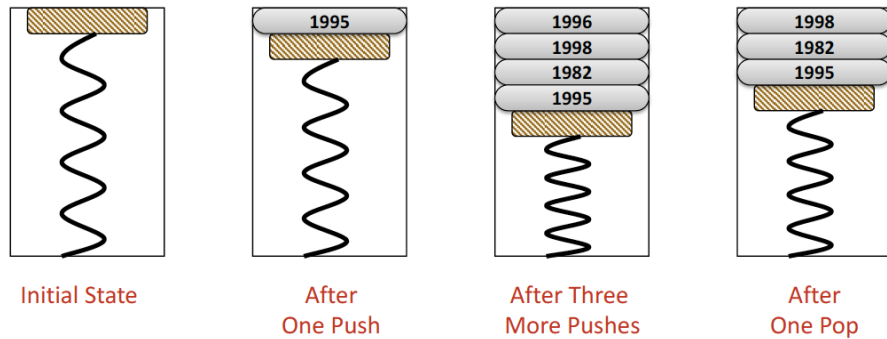
```
typedef struct studentStruct Node;  
struct studentStruct {  
    int UIN;  
    float GPA;  
    Node *prev;  
    Node *next;  
};
```



Stack & Queue abstract data types

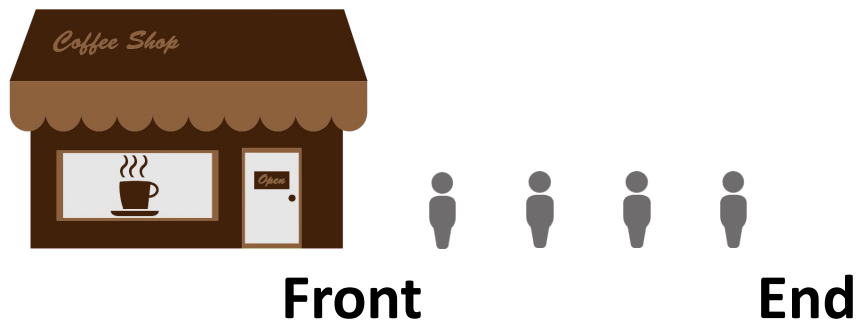
Stack

- First item in is the last item out - _____
- Two operations for data movement: _____ & _____



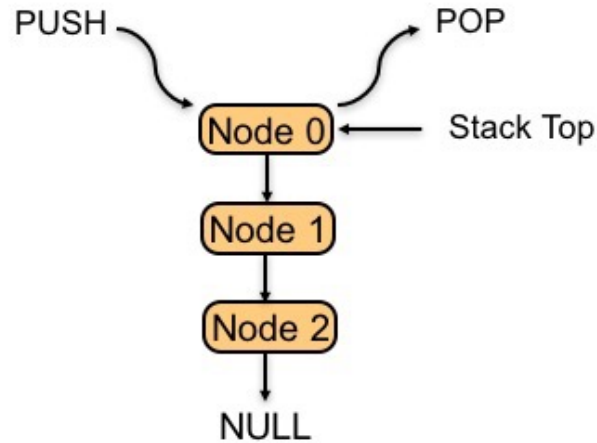
Queue

- First item in is the first item out - _____
- Two operations for data movement: _____ & _____

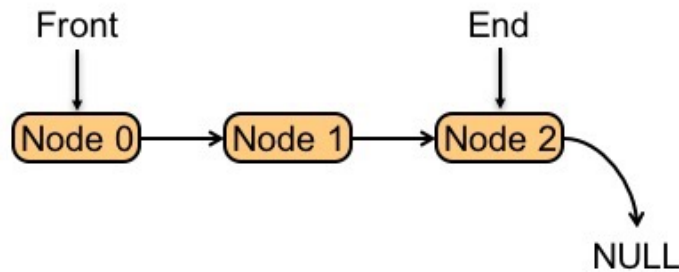


Implement abstract data types using linked list

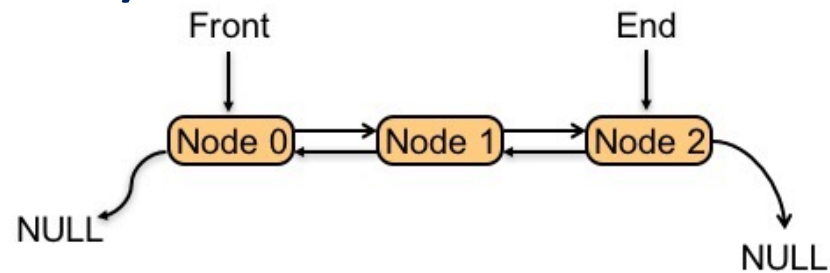
- Stack



- Queue



- Deque ("Deck", double-ended queue)



Stack implementation using a singly linked list

```
typedef struct nodeStruct node;
struct nodeStruct{
    int data;
    node *next;
};
node *top; /* global variable, init to NULL in main */
void push(int new_data); /* push a new node to stack */
int pop(); /* return data of the node pop from stack */
```