

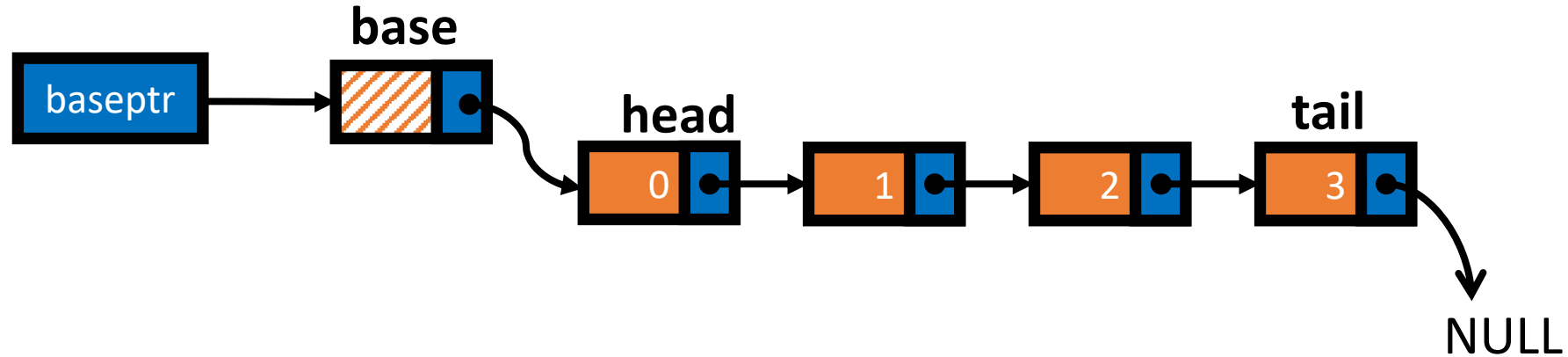
ECE 220: Computer Systems & Programming

Lecture 20: Problem Solving with Linked List Thomas Moon

March 26, 2024



Another way for head pointer



```
int main(){
    node base;
    base.next = NULL;

    node data;
    data.UIN = 1;
    ...

    insert_head_base(&base, &data);
```

```
void insert_head_base(node *baseptr, node
*data){
    node *temp = (node*) malloc(sizeof(node));
    *temp = *data;
    temp->next = baseptr->next;
    baseptr->next = temp;
}
```

```
int main(){
    node base;
    base.next = NULL;

    node data;
    data.UIN = 1;
    ...

    insert_head_base(&base, &data);
}
```

```
void insert_head_base(node *baseptr, node
*data){
    node *temp = (node*) malloc(sizeof(node));
    *temp = *data;
    temp->next = baseptr->next;
    baseptr->next = temp;
}
```

VS

```
int main(){
    node *headptr = NULL;

    node data;
    data.UIN = 0;
    ...

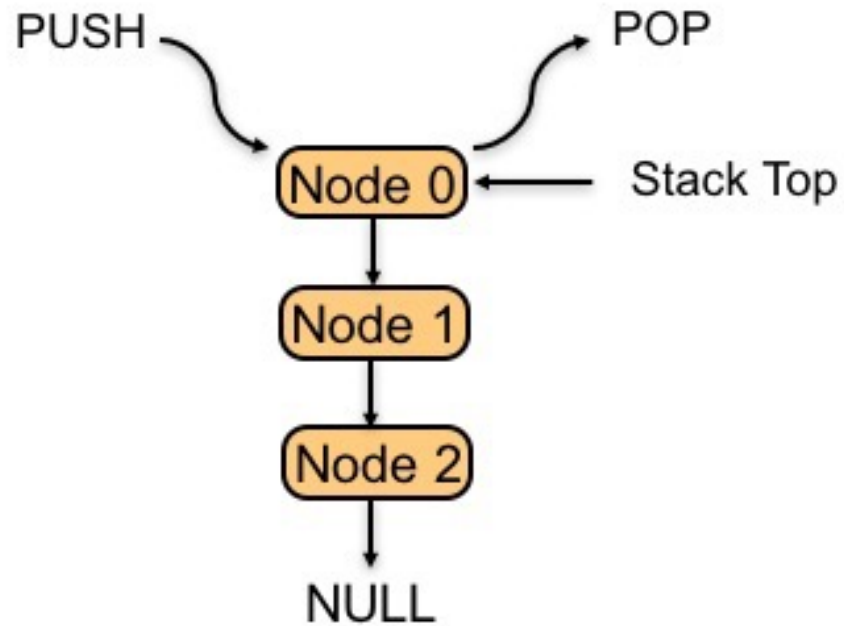
    insert_head(&headptr, &data);
}
```

```
void insert_head(node **headpptr, node *data){
    node *temp = (node*) malloc(sizeof(node));
    *temp = *data;
    temp->next = *headpptr;
    *headpptr = temp;
}
```

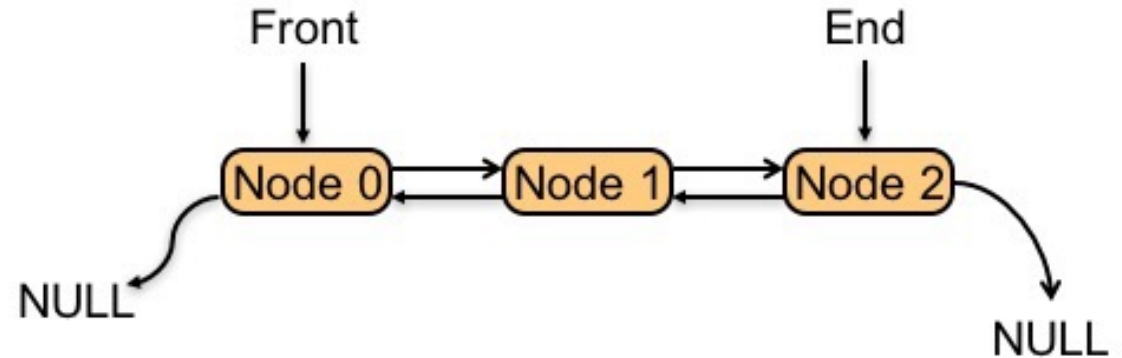
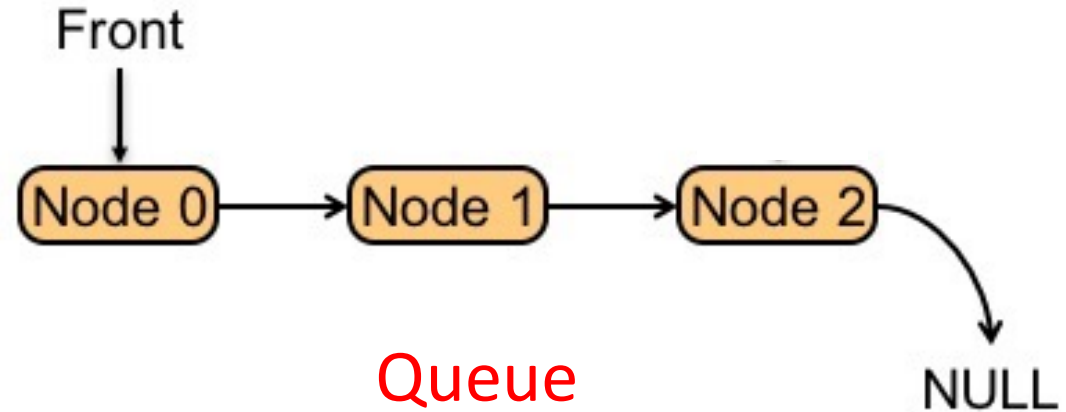
Tips how to implement linked-list (and tree) functions

1. Decide the argument for nodes: **single pointer** vs **double pointer**
 - **double ptr** if change the structure (add or remove nodes)
 - **single ptr** if read-only (print/search)
 - other variants possible
2. Traverse
 - while loop vs recursion
3. Connect/delete a node in a proper way
 - Be careful nothing lost

Implement Stack/Queue Using Linked List

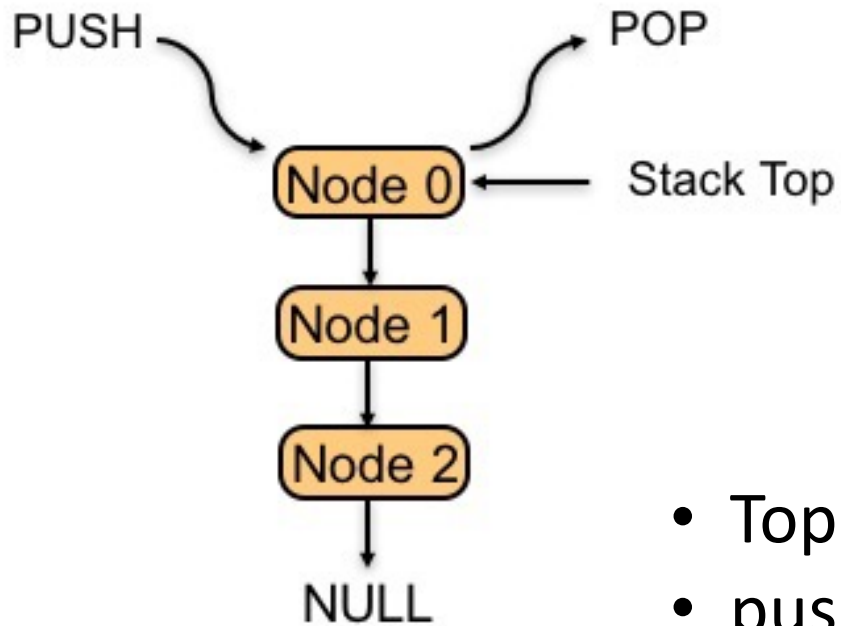


Stack



Deque("Deck", double-ended queue)

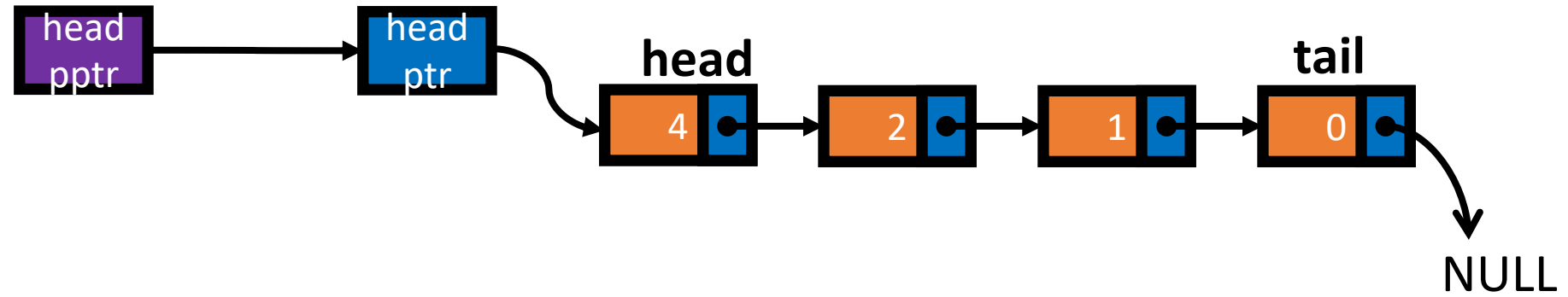
Stack by Linked List



Stack

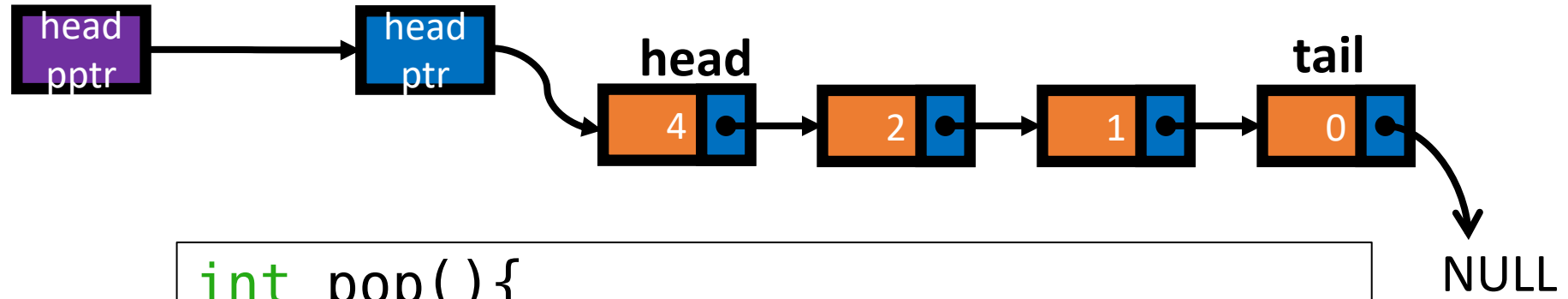
- Top pointer (=head pointer)
- push(): Insert a new node at the top.
- pop(): Remove the top.

Push for Stack



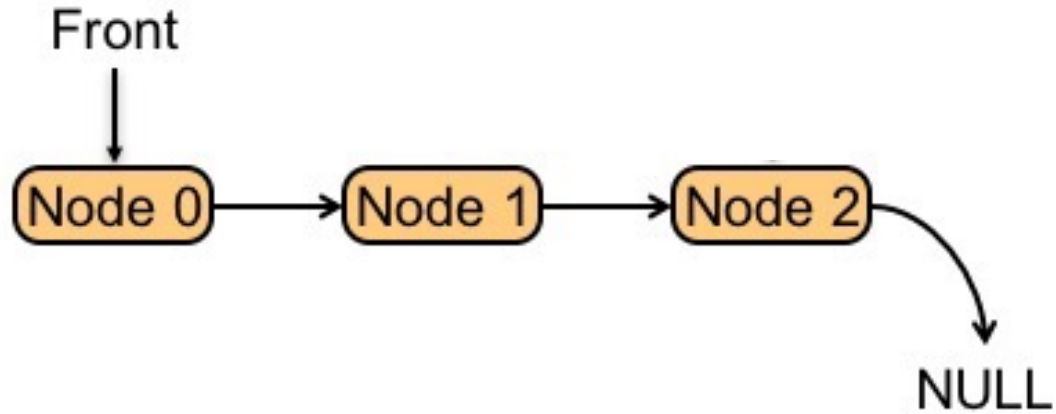
```
void push(){  
}
```

Pop for Stack



```
int pop(){  
}
```

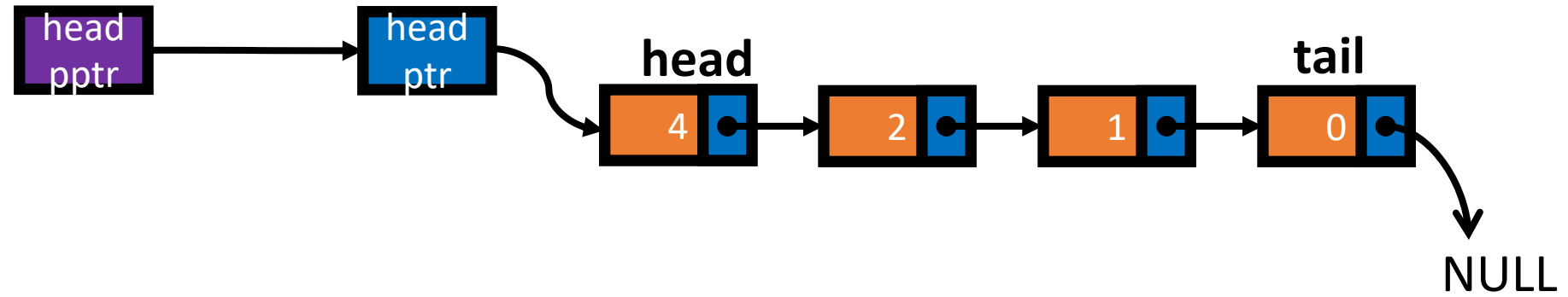

Queue by Linked List



Queue

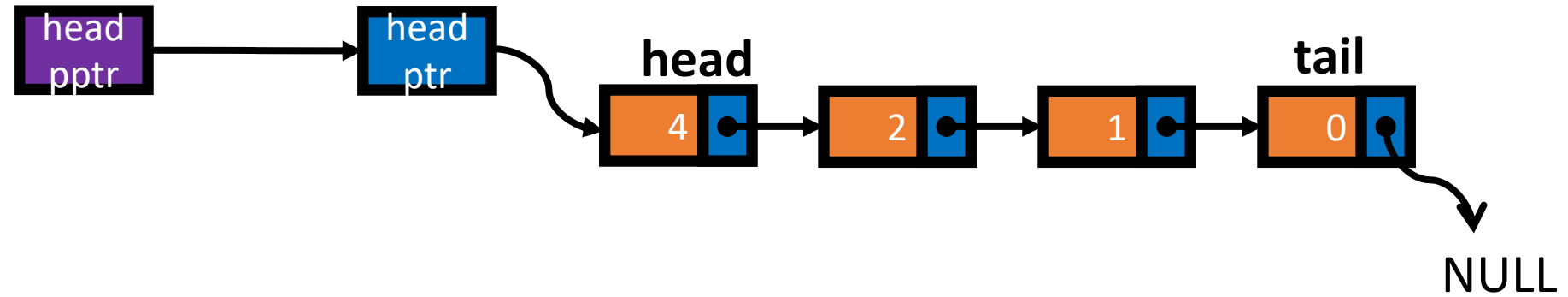
- Front pointer (=head pointer)
- enqueue(): Add a new node after the tail.
- dequeue(): Remove the head (**same as pop in stack**).

Dequeue for Queue



```
int dequeue() {  
}
```

Enqueue for Queue



```
void enqueue(){  
}
```

Midterm2 Review: 2 C programming, 1 C-to-LC3, 1 concept

- Basic grammar & concept on C
 - array, pointer, file I/O, structure
- Arrays
 - How to access 2D array element? How to pass 2D array arguments?
 - 1D array <-> 2D array (row major)
- Recursion
 - base case and recursive case
 - recursion with backtracking
- Run-time stack
 - Practice how to draw activation record on run-time stack
 - C to LC-3 – caller/callee build-up & tear-down
- We assume you DID MP's.