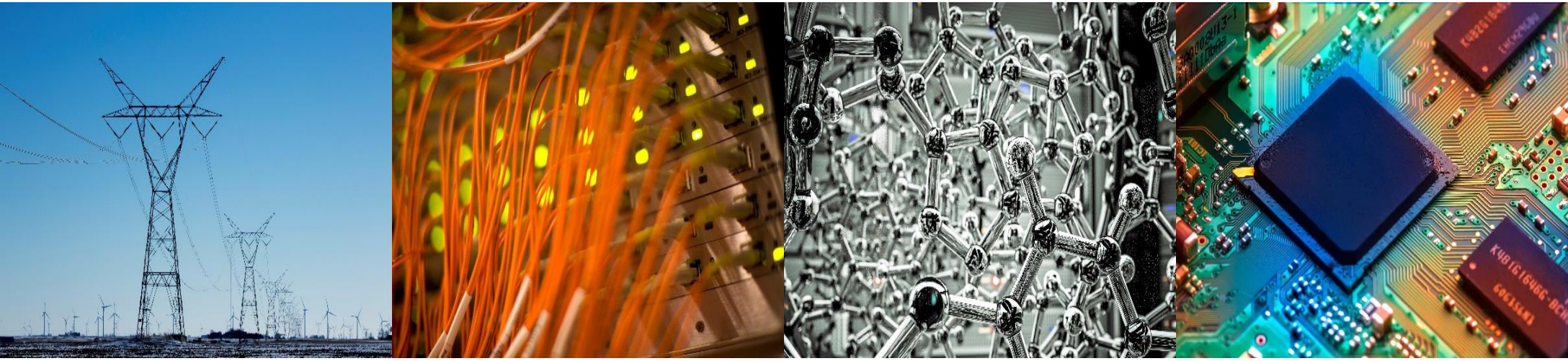
ECE 220 Computer Systems & Programming

Lecture 19 – C to LC-3 with Linked Data Structure

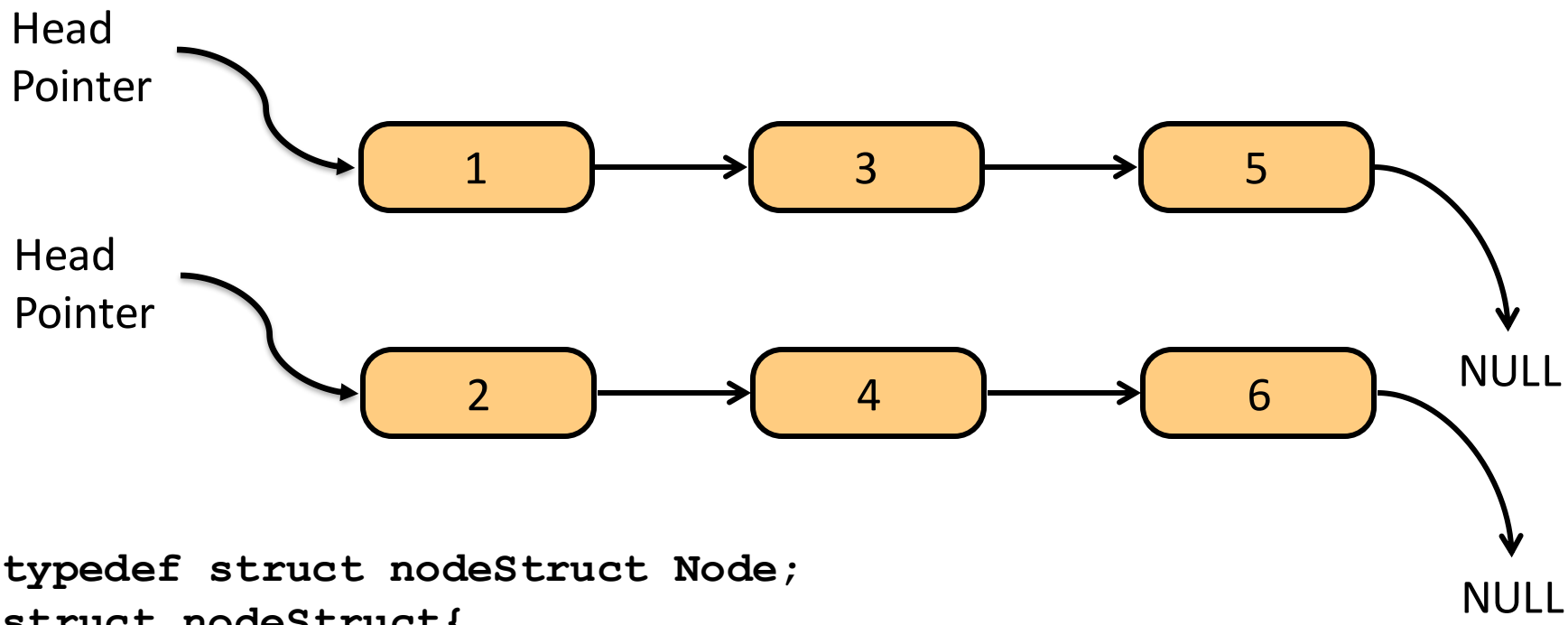
November 5, 2024



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

Exercise: Linked List

- Given two sorted linked lists of integers, merge them as one sorted linked list.



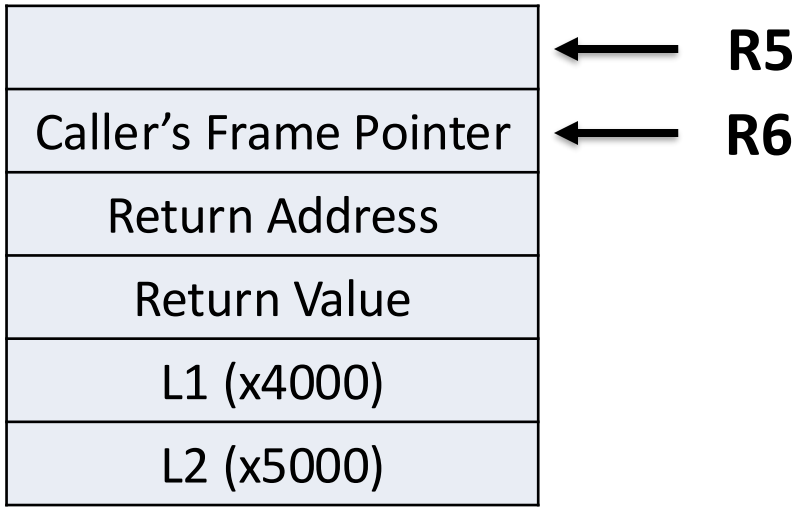

```
typedef struct nodeStruct Node;
struct nodeStruct{
    int data;
    Node *next;
};
```

Recursive Implementation

```
Node *mergeList(Node *L1, Node *L2){
   /* base cases: either L1 or L2 is NULL */



   /* recursive cases:
      (1) if data of head node in L1 is less than that in L2, set
      the next pointer of head node in L1 to point to the
      remaining merged list;(2) otherwise */

}
```

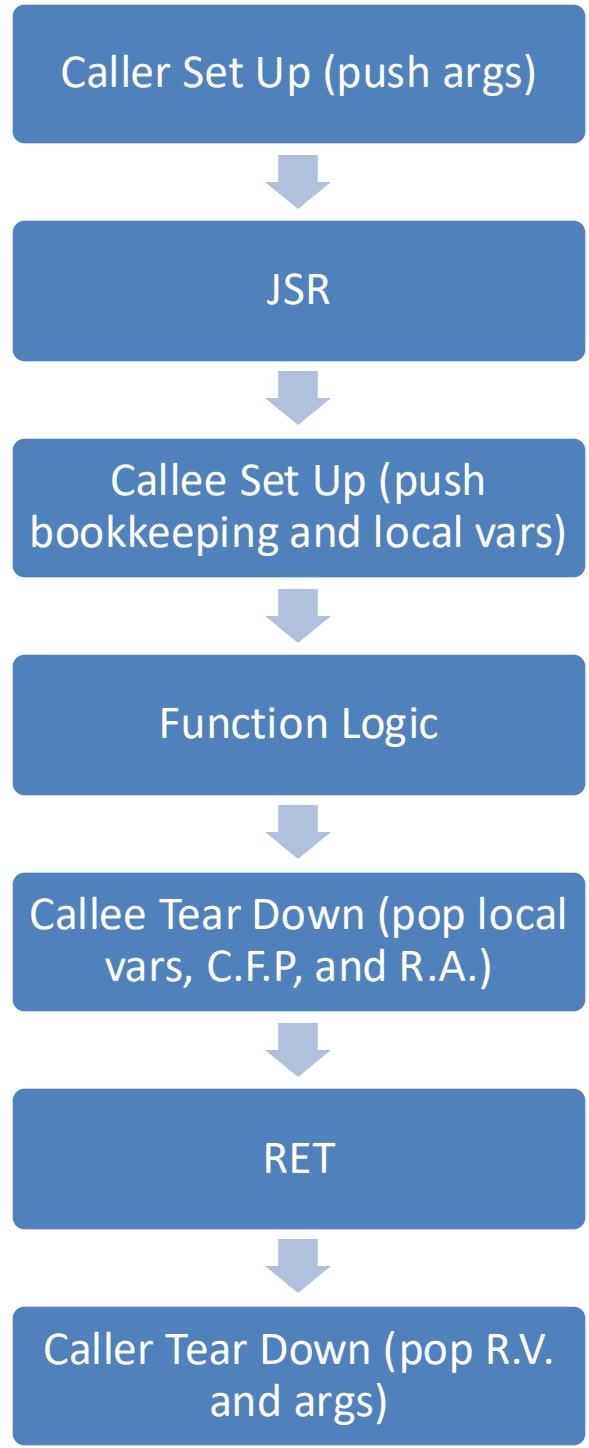
C to LC-3 Conversion

Run-Time Stack



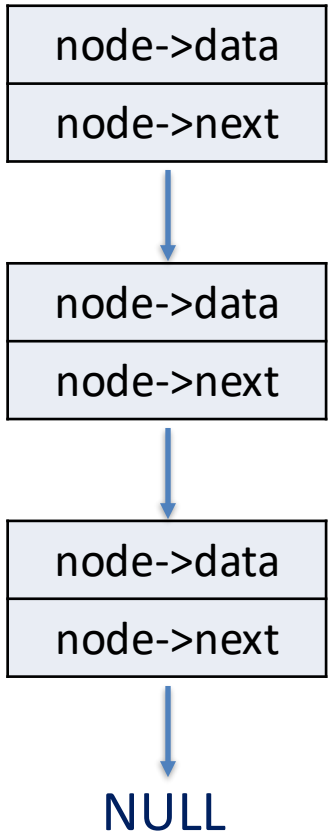
Heap

x4000	L1->data
x4001	L1->next



C to LC-3 Conversion

L1 starts at x4000, L2 starts at x5000



R1: L1 **R3**: L1->data / L1->next

R2: L2 **R4**: L2->data / L2->next

```
MERGE_LIST
;;Part 1 - callee build up
;allocate space for bookkeeping info


;store return address to stack


;store caller's frame pointer


;set up new frame pointer


;;Part 2 - implement function logic
;if(L1 == NULL)



;if(L2 == NULL)
```

```
;if(L1->data < L2->data)
;get L1->data and L2->data




;calculate L1->data - L2->data




;caller build up for mergelist(L1->next, L2)
```

```
;caller tear down for mergelist(L1->next, L2)
```

```
ELSE
;else
;caller build up for mergelist(L1, L2->next)

;caller tear down for mergelist(L1, L2->next)
```

```
RETURN_L1



RETURN_L2



;;Part 3 - callee tear down
DONE
```