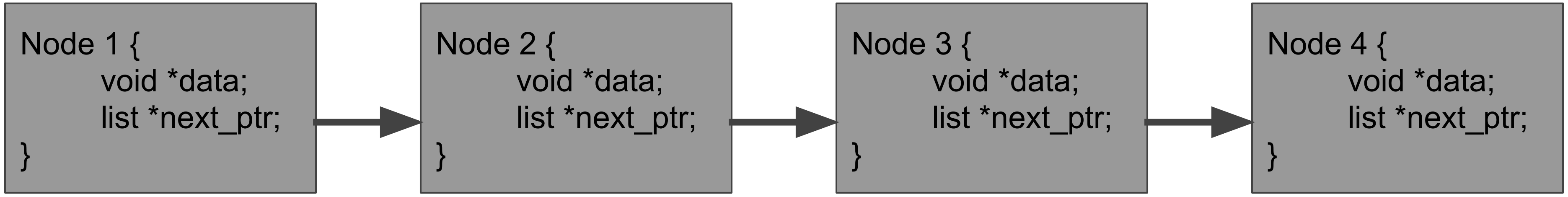
CS 24000 L04
Week 10

Extensions of Linked Lists, Malloc Part 2

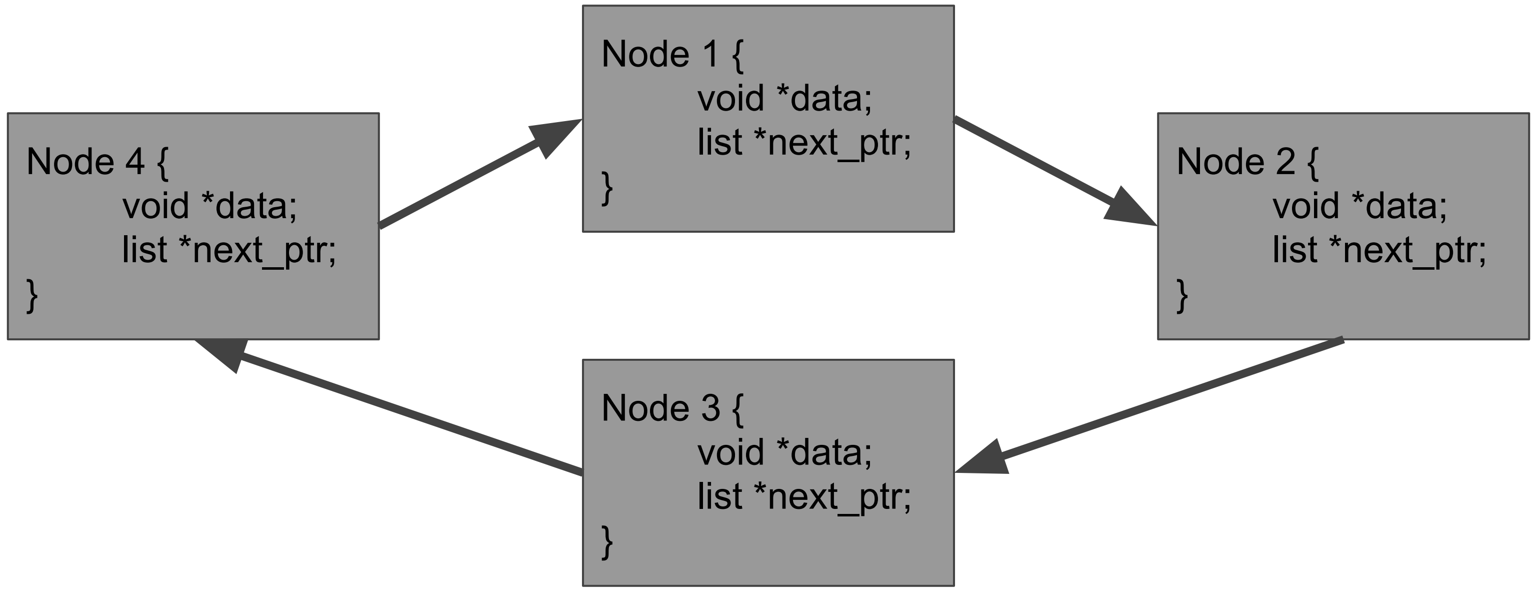
Linked Lists (Review)



- Every node has a pointer to the next node
- The last node typically points to NULL
    - BUT it doesn’t have to

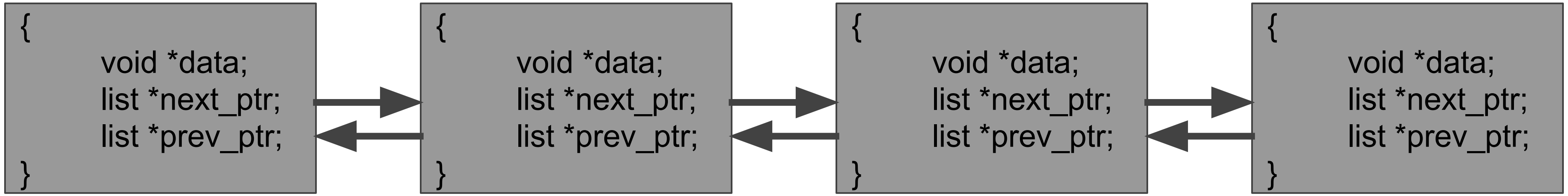
Circular/Cyclical Linked Lists

- An extension to linked lists
- The last node links back to the first node

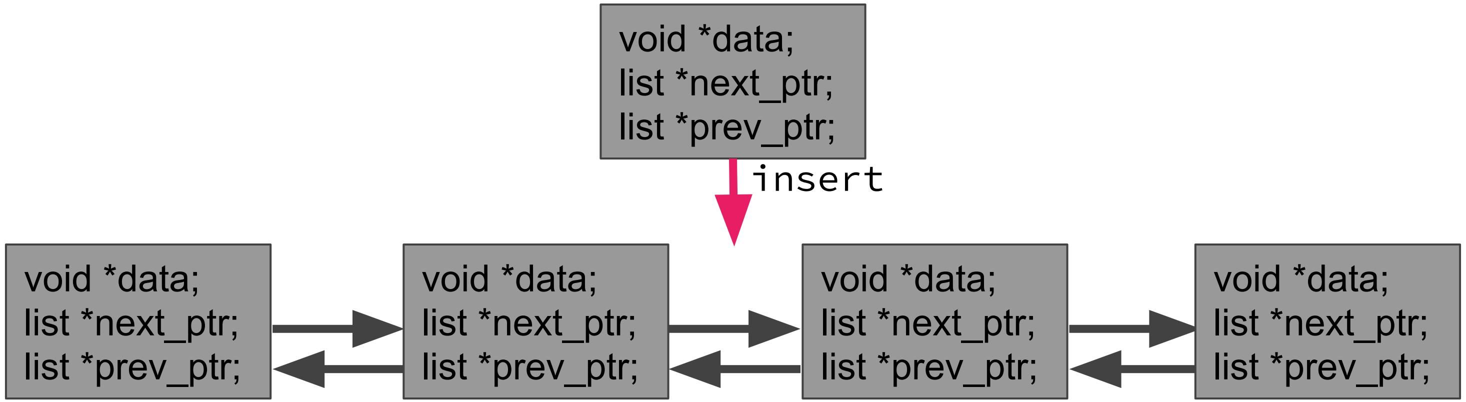


HW9 - Doubly Linked Lists (DLL)

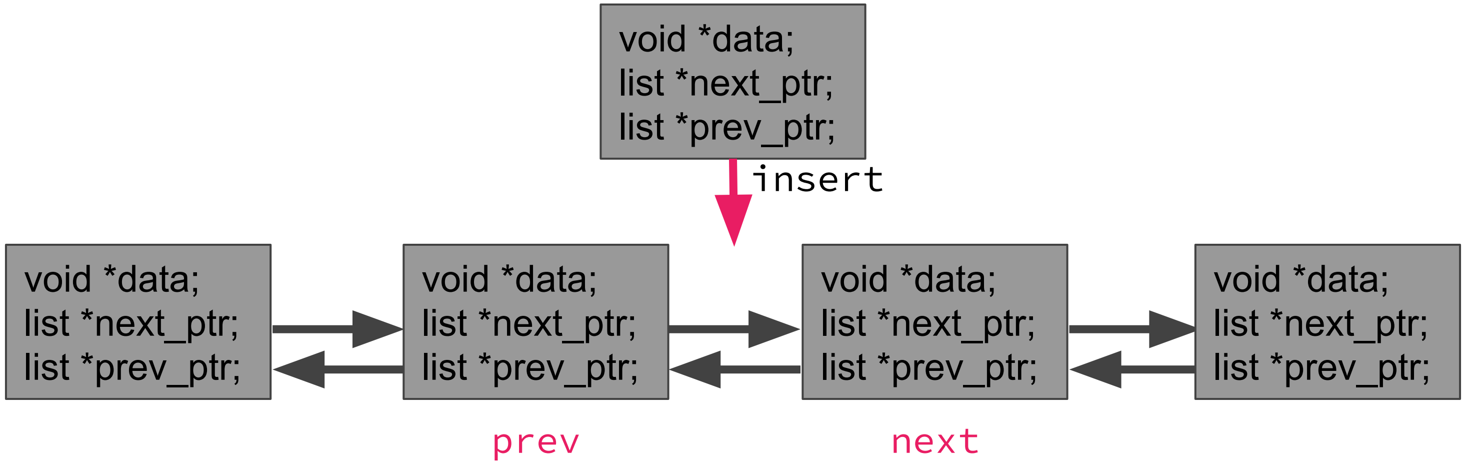
- Another extension to linked lists
- Every node has a next pointer and a previous pointer
- Traversal is possible in two directions



Insertion into a DLL

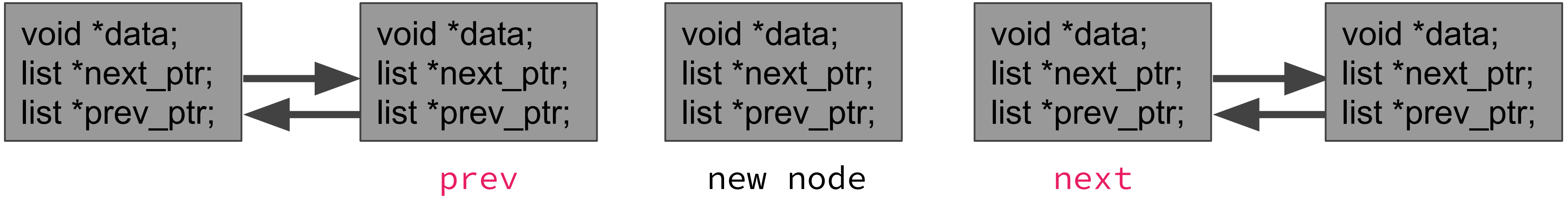


Insertion into a DLL



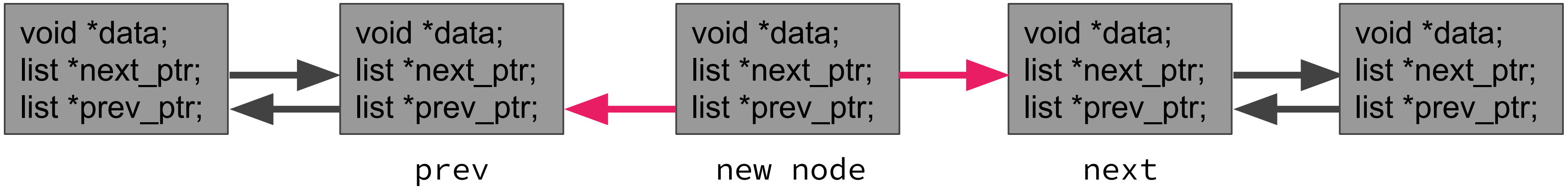
1) Store the previous and next pointers

Insertion into a DLL



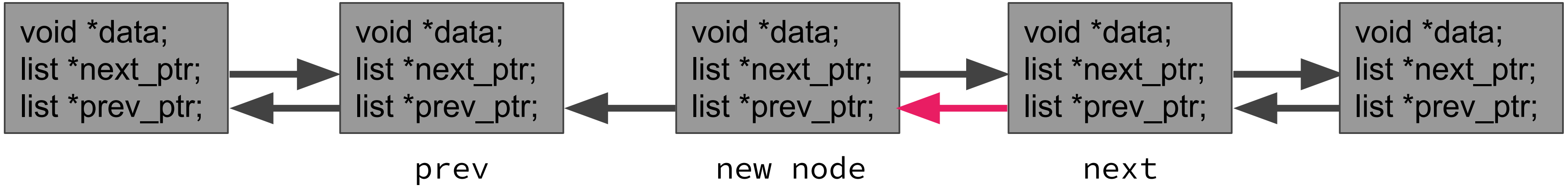
1) Store the previous and next pointers

Insertion into a DLL



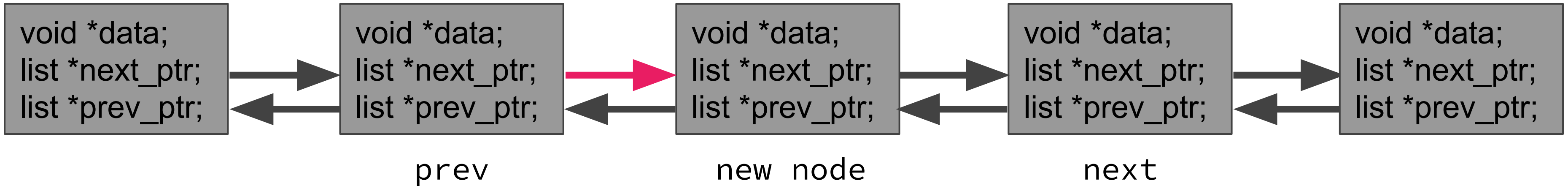
1) Store the previous and next pointers
2) Set the prev and next ptr of the new node

Insertion into a DLL



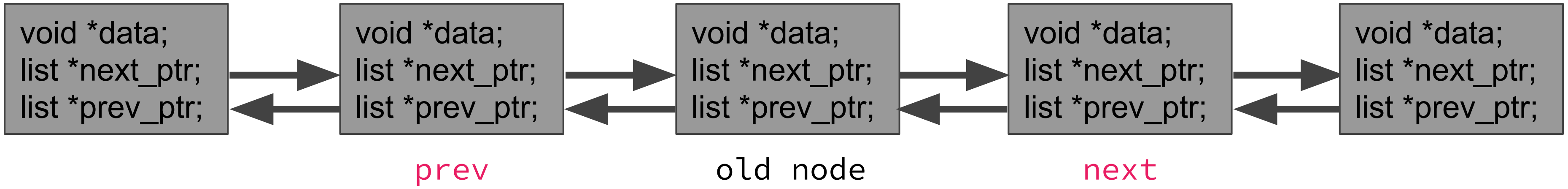
1) Store the previous and next pointers
2) Set the prev and next ptr of the new node
3) Set the prev of the next node to the new node

Insertion into a DLL



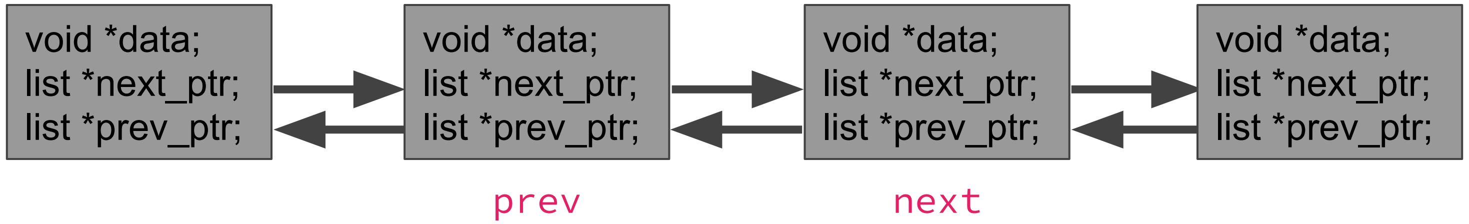
```
1)  Store the previous and next pointers
2)  Set the prev and next ptr of the new node
3)  Set the prev of the next node to the new node
4)  Set the next of the prev node to the new node
```

Deletion from a DLL



```
1)  Store the next_ptr and prev_ptr
2)  Set the next_ptr of prev to next
3)  Set the prev_ptr of next to prev
```

Deletion from a DLL



1) Store the next_ptr and prev_ptr as next and prev
2) Set the next_ptr of prev to next
3) Set the prev_ptr of next to prev

Quick Note on Malloc

- ALWAYS you will have as many mallocs in your program as frees
  - These have to be one-to-one
  - If you have add and delete functions, add will call malloc as many times as delete calls free
  - If a function adds and deletes, they will be in equal amounts
- If you ever malloc more memory than you free, that is a memory leak, and the HW tester will take off points

When you shouldn’t malloc

- If the memory already exists
  - Ex: In a linked list traversal, you don’t need to malloc anything
    - Just follow the pointers!
- For a temporary variable
  - If it doesn’t need to exist after the end of a function
    - Especially if it has a fixed size
- Generally speaking, don’t use malloc when you don’t have to