

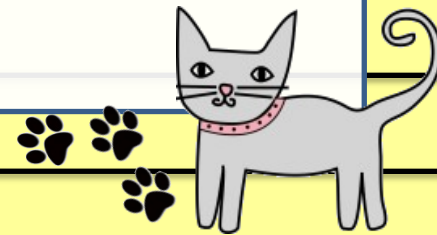
CMSC 21

Fundamentals of Programming

2nd Semester 2011-2012

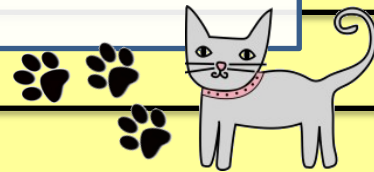


Circular Doubly Linked List

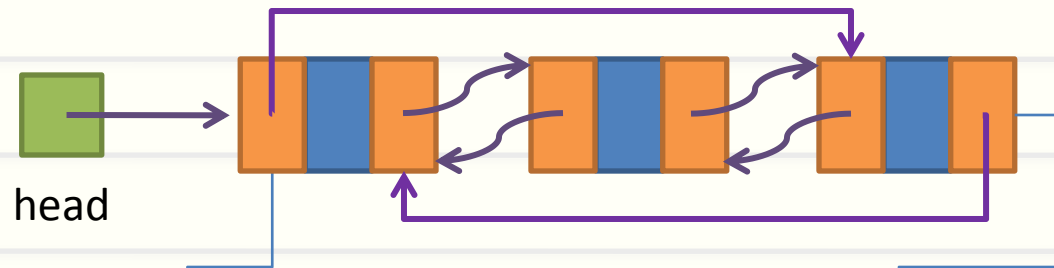


Circular Doubly Linked List

- The structure is similar to the doubly linked list, it has two pointer, a pointer to the previous node and a pointer to the next node
- No pointer in the linked list is set to NULL
- The list can be traversed clockwise or counter-clockwise

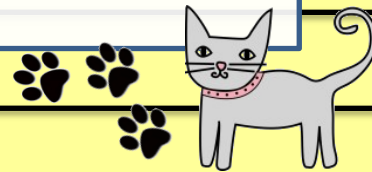


Circular Double Linked List



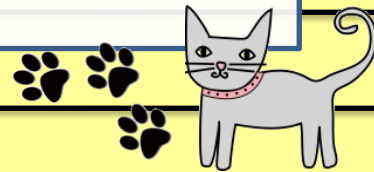
The previous pointer of the 'first node' is set to refer to the 'last node'

The next pointer of the 'last node' is set to refer to the 'first node'

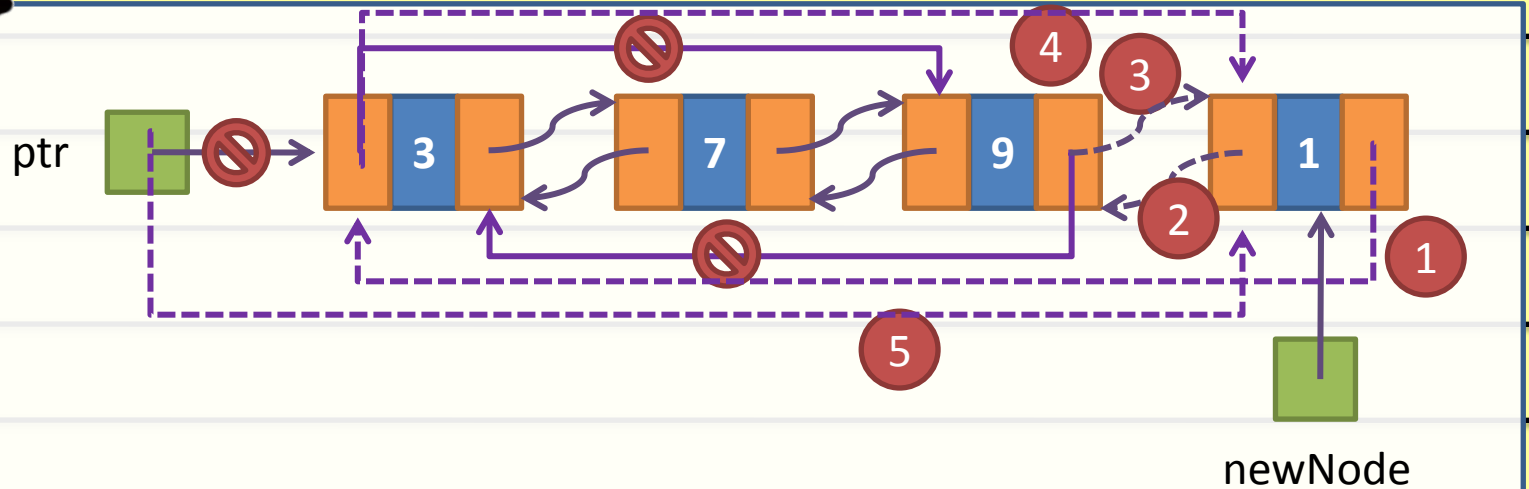


Operations

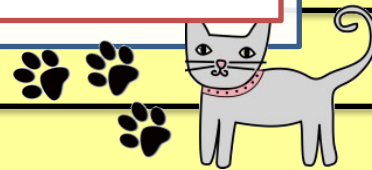
- Insert before the current node
- Insert after the current node
- Delete current node



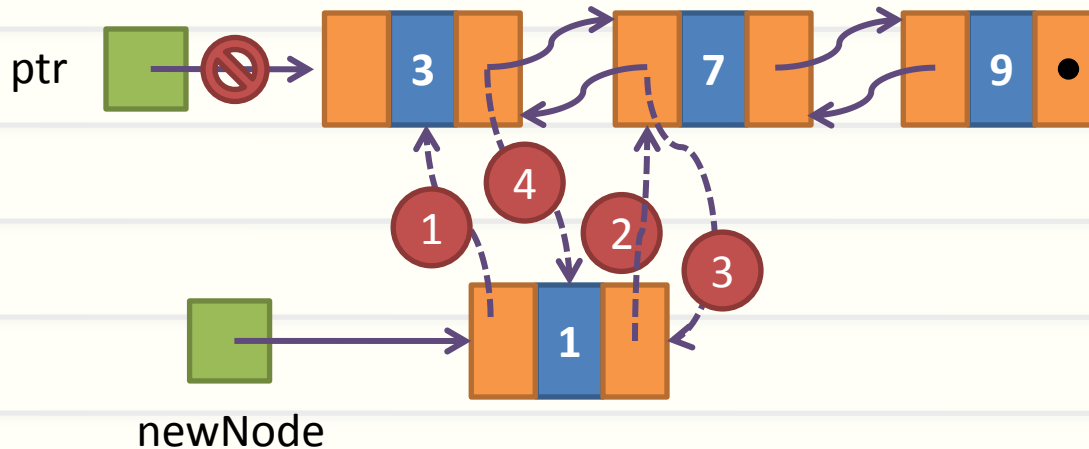
Insert before



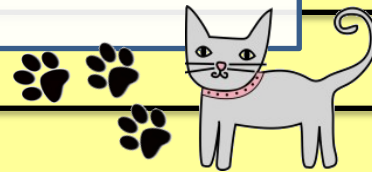
1. Make the next pointer of the new node point to the current node
2. Make the previous pointer of the new node refer to the node before the current node
3. Make the next pointer of the node before the current node point to the new node
4. Make the previous pointer of the current node point to the new node
5. Move ptr to refer to the new node



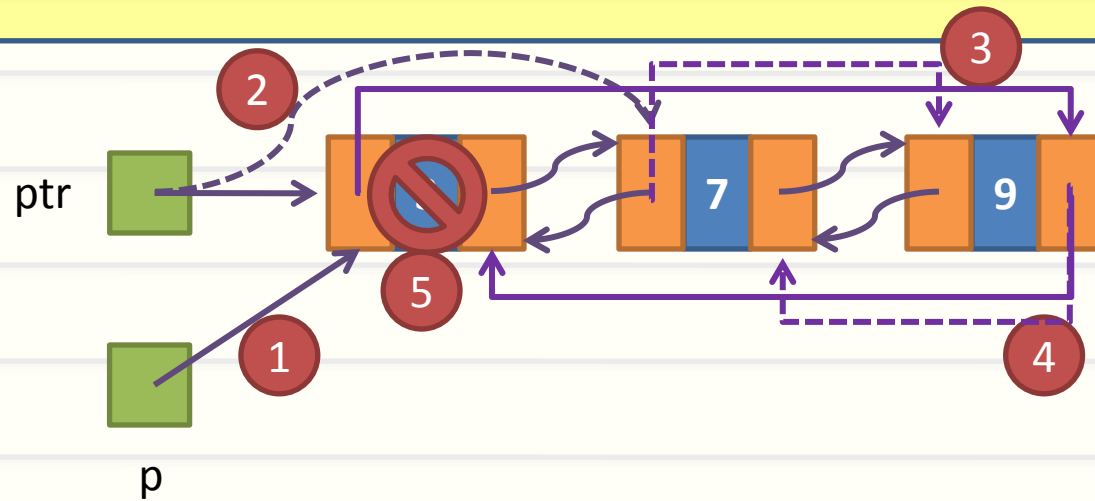
Insert after



1. Make the previous pointer of the new node refer to the current node
2. Make the next pointer of the new node refer to the node next to the current node
3. Make the previous pointer of the node next to the current node refer to the new node
4. Make the next pointer of the current node refer to the new node



Delete current node



1. Have another pointer refer to the current node
2. Move ptr to the next node
3. Make the previous pointer from step 2 refer to p->prev
4. Make the next pointer of p->prev refer to the node in step 2
5. Free the node referred to by p

