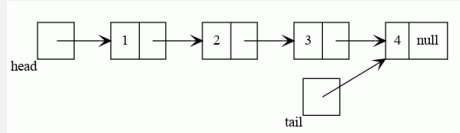


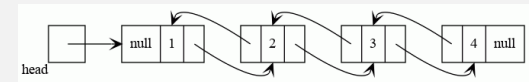
Variations on Linked Lists

- tail pointers
 - speed up access/insert at end by maintaining both head and tail pointers



Variations on Linked Lists

- doubly-linked list
 - speed up insert, remove, remove at tail – anything that involves a node before – by having node store next and prev pointers



Variations on Linked Lists

- circular list
 - tail's next points back to the head instead of being null
 - can reduce special cases for handling first and last nodes
 - get the benefits of a tail pointer with only a single pointer
 - convenient for round-robin scheduling

