Problem: Implement a queue with enqueue and dequeue methods using a linked list.

Constraints

- **If there is one item in the list, do you expect the head and tail pointers to both point to it?**
    - **Yes**
- **If there are no items on the list, do you expect the head and tail pointers to be None?**
    - **Yes**
- **If you dequeue on an empty queue, does that return None?**
    - **Yes**
- **Can we assume this fits memory?**
    - **Yes**

Test Cases

Enqueue

- **Enqueue to an empty queue**
- **Enqueue to a non-empty queue**

Dequeue

- **Dequeue an empty queue -> None**
- **Dequeue a queue with one element**
- **Dequeue a queue with more than one element**

Algorithm

Enqueue

- **If the list is empty, set head and tail to node**
- **Else, set tail to node**

Complexity:

- **Time: O(1)**
- **Space: O(1)**

Dequeue

- **If the list is empty, return None**
- **If the list has one node**
  - **Save the head node's value**
  - **Set head and tail to None**
  - **Return the saved value**
- **Else**
  - **Save the head node's value**
  - **Set head to its next node**
  - **Return the saved value**

Complexity:

- **Time: O(1)**
- **Space: O(1)**