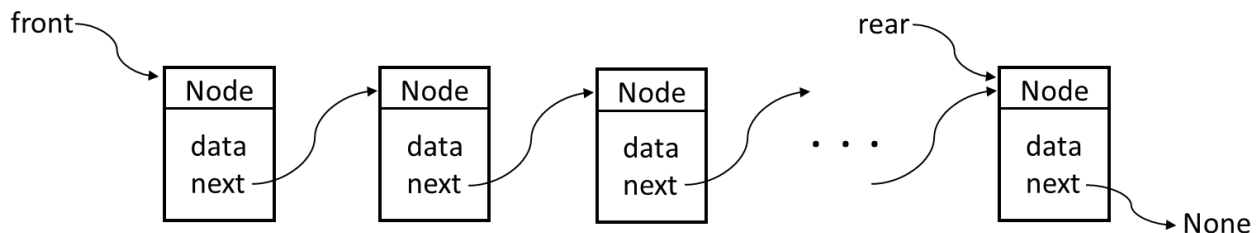# Lab 3: Queue Implementations

Goals:

- **Implement a Queue class using a link-based data structure: queue_linked.py**
- **Implement a Queue class using a circular array: queue_array.py**

Before doing this lab make sure to read over section 3.10, 3.11, and 3.12 carefully. You will not be doing the implementation described in section 3.12, but it will help you understand the underlying concepts.

- In the first implementation, you will use a linked structure similar to the linked structure used in implementing the Stack ADT (i.e. create a Node class). In this case, there must be a way to add items to the rear of the list and remove items from the front of the list, as illustrated below.

- The second implementation will use a circular array for storing the items in the queue. There are different ways of doing this, but they share the idea presented in the picture below. Items are added to the rear of the queue using the next "free entry" in an array. Items are removed from the front of the queue. The indices front and rear are incremented as items are added and deleted from the queue, and the indices "wrap around" when they reach the end of the array. Why is using a circular array better than what the text does (i.e. inserting new items at index 0 of array, and removing items from the end of the array)?

CPE 202
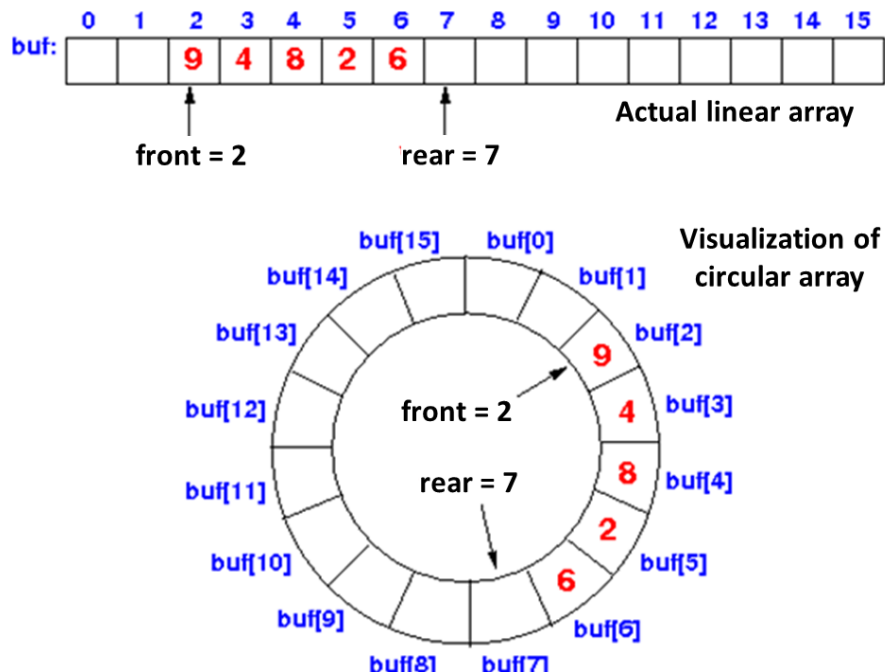
For both Queue implementations:

- Attempting to enqueue an item into a full Queue will raise an IndexError
- Attempting to dequeue an item from an empty Queue will raise an IndexError
- All methods must have *O(1)* performance (the speed must not be affected by the size of the queue)
- As with your stack implementations, you **may NOT use** any of the following Python List operations:
  - `append()`
  - `insert()`
  - `extend()`
  - `remove()`
  - `pop()`
  - + (concatenations)
  - List slicing (e.g. `some_list[2:9]`)

The following starter files are available in PolyLearn. Complete the implementations and ensure that your implementations are submitted to PolyLearn.

- **`queue_array.py`**: Contains an array (Python List) based implementation of the **Queue** class
- **`queue_linked.py`**: Contains a linked based implementation of the **Queue** class
- **`queue_array_tests.py`:** Contains your set of thorough tests to ensure your implementations work correctly for the array implementation.
- **`queue_linked_tests.py`:** Contains your set of thorough tests to ensure your implementations work correctly for the linked list implementation.