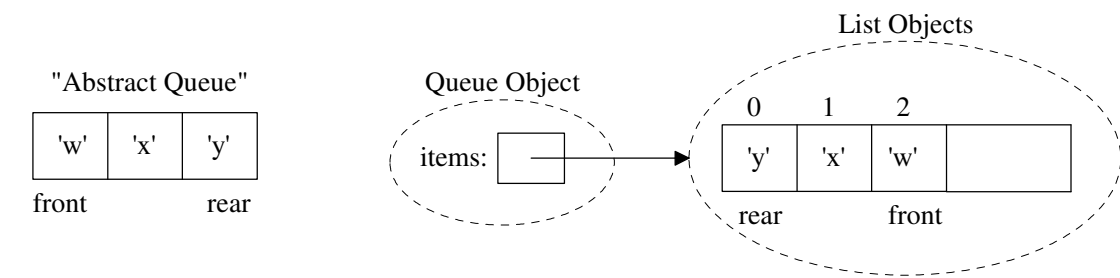


Objective: To understand FIFO (First-In-First-Out) queue implementations in Python including being able to determine the big-oh of each operation.

To start the lab: Download and unzip the file at: www.cs.uni.edu/~fienup/cs1520s13/labs/lab3.zip

Part A: The textbook’s QueueText implementation in lab3/queue_text.py uses a Python list



a) Complete the big-oh notation for the above QueueText implementation: ("n" is the # items)

	__init__	enqueue(item)	dequeue()	peek()	size()	isEmpty()	__str__
Big-oh							

b) Explain your big-oh answer for enqueue(item).

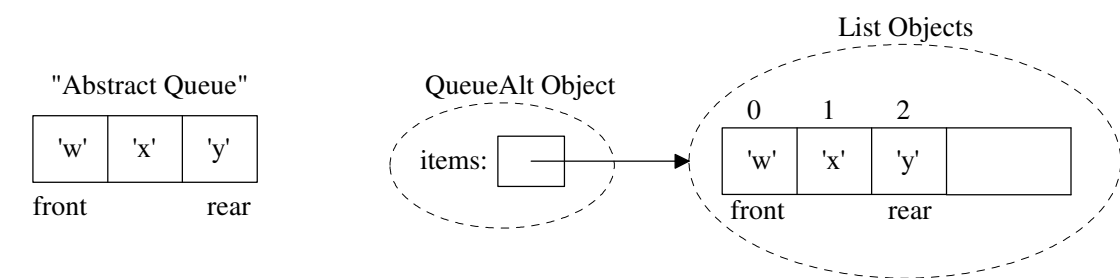
c) Explain your big-oh answer for dequeue()

d) Run the timeQueue.py file which times 100,000 enqueues followed by 100,000 dequeues.
Time for 100,000 enqueues: _____ Time for 100,000 dequeues: _____

e) Why do the enqueues take so much more time?

After answering the above questions, raise you hand and explain your answers.

Part B: Complete the QueueAlt implementation in lab3/queue_alt.py uses a Python list



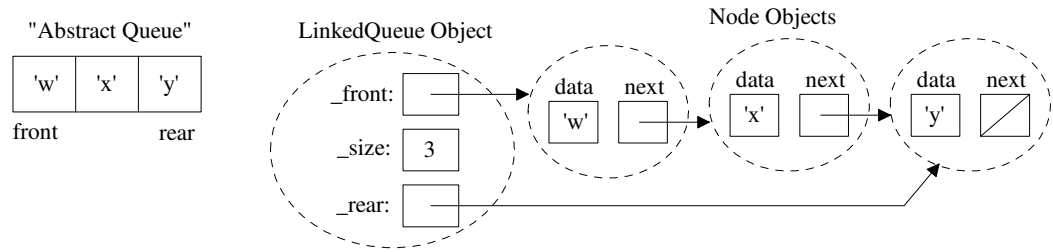
a) Complete the big-oh notation for the above QueueAlt implementation: ("n" is the # items)

	__init__	enqueue(item)	dequeue()	peek()	size()	isEmpty()	__str__
Big-oh							

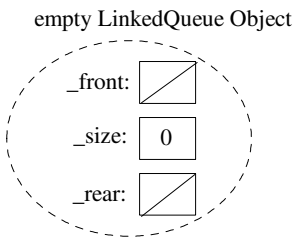
b) Run the timeQueueAlt.py file which times 100,000 enqueues followed by 100,000 dequeues.
Time for 100,000 enqueues: _____ Time for 100,000 dequeues: _____

After completing the QueueAlt class, answering the above questions, raise you hand and demonstrate your code.

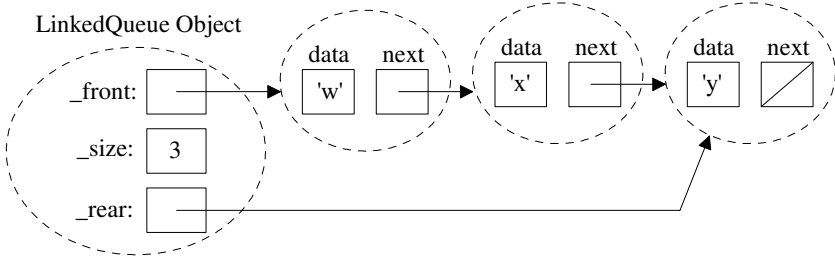
Part C: Consider the `LinkedList` implementation in `lab3/linked_queue.py` which uses a linked structure that looks like:



- a) Draw the picture and number the steps for the `enqueue` method of the “normal” case (non-empty queue) above
- b) Complete the `enqueue` method code for the “normal” case in the `lab3/linked_queue.py` file
- c) Starting with the empty queue below, draw the resulting picture after your “normal” case code executes.



- d) Fix your “normal” case code to handle the “special case” of an empty queue.



- e) Draw the picture and number the steps for the `dequeue` method of the “normal” case (non-empty queue) above
- f) Complete the `dequeue` method code for the “normal” case in the `lab3/linked_queue.py` file
- g) What “special case(s)” does the `dequeue` method code need to handle?

h) Complete the big-oh notation for the `LinkedList` methods: ("n" is the # items)

	<code>__init__</code>	<code>enqueue(item)</code>	<code>dequeue()</code>	<code>peek()</code>	<code>size()</code>	<code>isEmpty()</code>	<code>__str__</code>
Big-oh							

- i) Run the `timeLinkedList.py` file which times 100,000 enqueues followed by 100,000 dequeues.
Time for 100,000 enqueues: _____ Time for 100,000 dequeues: _____

After thoroughly testing your linked implementation, raise you hand and demonstrate your queue.