Queues

- Queues are quite similar to stacks
  - the difference is that the first item places, is the first to be removed
- Characteristics
  - Items come off the queue in the same order they went into it
    - FIFO (First in First out)

Queue ADT

- The Queue ADT:
  - is_empty() # is the queue empty
  - enqueue() # add item to the queue
  - dequeue() # remove item from the queue
- Queue can be used as follows:

```
queue = Queue()
for v in "aeiou":
    queue.enqueue(v)

while not queue.is_empty():
    print(queue.dequeue())
```

this will add the vowels to the queue in order, then print them back out in the same order

---

Class Definition

```
class Queue:
    def __init__(self, capacity=10):
        self.data = [0] * capacity
        self.front = 0
        self.back = 0

    def count(self):
        if self.front >= self.back:
            return self.front - self.back
        else:
            return self.front - self.back + len(self.data)

    def is_empty(self):
        return slef.front == self.back
```

```
def enqueue(self, item):
    if self.count() < len(self.data) - 1:
        self.data[self.front] = item
        self.front = (self.front + 1) % len(self.data)
    else:
        raise Exception("Queue Full")

def dequeue(self):
    item = self.data[self.back]
    self.back = (self.back + 1) % len(self.data)
    return item
```

Complexity Analysis

Operation	Big O Performance
is_empty()	Constant Time
enqueue()	Constant Time
dequeue()	Constant Time
