

CS218 - Data Structures

FAST NUCES Peshawar Campus

Dr. Nauman (recluze.net)

September 15, 2019

1 Queue using “Array”

Raster images of the notebook 07-queue

Queue

Queue code is exactly the same as we did with C++.

```
In [ ]: class Queue:
        def __init__(self):
            self.size = 5
            self.q = list(range(self.size)) # some dummy values
            self.i = 0
            self.o = 0

            self.is_empty = True
            self.is_full = False
```

```
In [ ]: q = Queue()
```

```
In [ ]: def _inc(self, idx):
        if idx + 1 == self.size:
            return 0
        else:
            return idx + 1
        Queue._inc = _inc
```

```
In [ ]: def enqueue(self, val):
        if self.is_full:
            raise IndexError("Queue full. Cannot enqueue.")

        self.q[self.i] = val
        self.i = self._inc(self.i)

        if self.i == self.o:
            self.is_full = True

        self.is_empty = False

        Queue.enqueue = enqueue
```

```
In [ ]: def dequeue(self):
        if self.is_empty:
            raise IndexError("Queue empty. Cannot dequeue.")

        ret = self.q[self.o]
        self.o = self._inc(self.o)

        if self.i == self.o:
            self.is_empty = True

        self.is_full = False
        return ret

        Queue.dequeue = dequeue
```

```
In [ ]: def __str__(self):
        return str(self.q) + " , in: " + str(self.i) + " , out: " + str(self.o)
        Queue.__str__ = __str__
```

```
In [ ]: q = Queue()
q.enqueue(10)
q.enqueue(20)
q.enqueue(30)
q.enqueue(40)
q.enqueue(50)
# q.enqueue(35)

print(q)

print(q.dequeue())
q.enqueue(110)
print(q)

print(q.dequeue())
print(q.dequeue())
print(q.dequeue())
print(q.dequeue())
print(q.dequeue())
# print(q.dequeue())

print(q)
```