Data Structures

Circular Queue Code

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Circular Queue Code

- We can create an array like Array Section
- For simplicity, let's just use the list to create N initial fixed elements
- _next
  - Utility to move the a position one step consider the cycling

```
class Queue:
    def __init__(self, size):
        self.added_elements = 0
        self.rear = self.front = 0
        # Let's use list to express our FIXED array
        self.array = [None] * max(1, size)

    def _next(self, pos):
        pos += 1
        if pos == len(self.array):
            pos = 0
        return pos
    # return (pos + 1) % size   #  Or shorter way
```

How to check empty and full status?

- Trivially handled using added_elements

```
def empty(self):
    return self.added_elements == 0

def full(self):
    return self.added_elements == len(self.array)
```

Enqueue and Dequeue

- Enqueue: add to the rear index, and move rear
- Dequeue: get from the front index, and move front
- Maintain the added_elements variable

```
def enqueue(self, value):
    assert not self.full()
    self.array[self.rear] = value
    self.rear = self._next(self.rear)
    self.added_elements += 1

def dequeue(self):
    assert not self.empty()
    value = self.array[self.front]
    self.front = self._next(self.front)
    self.added_elements -= 1
    return value
```

Display Queue

- Simply start from the front and count based on added_elements

```
def display(self):
    print(f"Front {self.front} - rear {self.rear}", end = '\t')

    if self.full():
        print("FULL", end='')
    elif self.empty():
        print("EMPTY\n")
        return

    print("")
    cur = self.front

    for step in range(self.added_elements):
        print(self.array[cur], end=" ")
        cur = self._next(cur)
    print("")
```

Usage

- Please refer to the attached code
- Read the whole main and check the output

```
qu = Queue(6)

assert qu.empty()
qu.display()

for i in range(1, 7):
    assert not qu.full()
    qu.enqueue(i)
    qu.display()

print()

assert qu.full()
for i in range(1, 7):
    assert not qu.empty()
    qu.dequeue()
    qu.display()
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."