Queue Data Structure

Vivek K. S., Deepak G.

Information Systems Decision Sciences (ISDS)
MUMA College of Business
University of South Florida
Tampa, Florida

2017

What is a Queue

- A queue is an ordered collection of items.
- Data items are added through one end called the “rear” and removed through the other called the “front”.
- It follows the FIFO ordering principle also known as ”first-come first-served.”
- Ticket counters and Printing tasks in a library are real world examples of a queue.

Essential Operations in a Queue

- Enqueue - Ability to add a new item to the queue.
- Dequeue - Ability to remove an item from the queue.
- Ability to check if the queue is empty.
- Ability to check the size of the queue.

Logical Approach to Implementing a Queue

- We need to be able to create new queue instances on the go. Hence we will take an object oriented approach towards building a Queue and defining its behavior.
- The list data structure provides us with the methods to perform all the essential operations.
- Enqueuing can be done from the rear end using insert() method at index position 0 so the insertion operation will be a O(1) operation.
- Dequeuing can be done from the front end using the pop() method in lists.

Continued..

- The comparator operator will return a True/False based on the fact that the queue is empty or not.
- The len() method can be used to check the size of the queue.

Python code

Code Implementation in Python

```
class Queue:
        def __init__(self):
                self.items = []
        def is_empty(self):
                return self.items == []
        def add(self, item):
                self.items.insert(0,item)
        def remove(self):
                return self.items.pop()
        def length(self):
                return len(self.items)
```

Summary

- Queues are abstract data structures that can be built using a List type.
- It follows the FIFO ordering principle and finds its application in most real world applications that follow the FIFO ordering principle, for example a printing task in a lab.
- Queues are very useful in most computing applications and the right implementation gives the best performance, as seen in the enqueue operation which is O(1).