Deque 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 Deque

- A Deque is an ordered collection of items.
- It has two ends called the *front* and the *rear* which are interchangeable unlike the queue data structure.
- A deque is different from a stack or a queue.
- Data items can be added and removed from both the ends and thus it assumes the characteristics of both a stack and a queue.
- It could be thought of as a crossover between a stack and a queue.
- It does not follow any ordering principle such as LIFO or FIFO.

Essential Operations in a Deque

The following are the essential operations of a Deque.

- To add a new item either to the front or the rear end of the deque.
- To add or remove an item from both the front or the rear end of the deque.
- To check if the deque is empty.
- To find the size of the deque.

Logical Approach to Implementing a Deque

- 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 these operations.
- The ability to add and remove items from both the ends of the list is provided by the list indices.
- To add an element to the front, we can easily do a traditional list insert.
- To remove elements from the front, we use the traditional pop() method.
- This implementation of adding and removing items from the front is O(n) operation.
- This is because, we have to traverse all the way to the end (which is the front) to be able to do add and remove items from the "front".

Continued..

- Likewise, the removal of item from the rear can be done by using the pop() method at the index position 0.
- To add an element to the rear, we will be inserting the element to the index position 0.
- This implementation of adding and removing items from the rear is O(1) operation.
- Just like in queues we will use the len() and comparator operation to check the size and for empty deques.

Python code

Code Implementation in Python

```
class Deque:
        def __init__(self):
                self.items = []
        def is_empty(self):
                return self.items == []
        def add_to_front(self, item):
                self.items.append(item)
        def add_to_rear(self, item):
                self.items.insert(0,item)
        def remove_from_front(self):
                return self.items.pop()
        def remove_from_rear(self):
                return self.items.pop(0)
        def length(self):
                return len(self.items)
```

Summary

- Deques are Abstract data structures that could be thought of as a mix of both stacks and queues.
- Dequeus can be implemented with Python's in-built List data structure.
- Deques adopt both LIFO and FIFO ordering principles and can be chosen between based on the requirements.
- Deques have an O(1) operation for adding and removing items if the rear end is chosen to add and remove items.