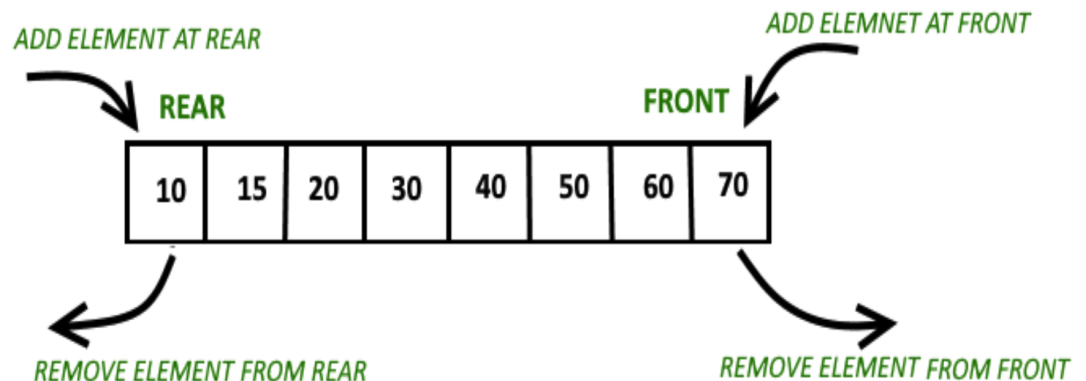


Deque (Doubled Ended Queue)

Data structure Team Research Assignment

Present/Report on:

- What is your data structure?
 - a hybrid queue where you can add and remove elements to and from both ends of the queue.



- Are there different ways to implement it? If so, what are they?
 - Doubly List
 - contains an extra pointer, typically called *previous pointer*, together with next pointer and data which are there in singly linked list.
 - Circular Array
 - For implementing deque, we need to keep track of two indices, front and rear. We enqueue(push) an item at the rear or the front end of dequeue and dequeue(pop) an item from both rear and front end.
 -
 - Add elements from either side
 - Using the dq.add command
 - Using the dq.addFirst("")
 - Adding elements on the left side

- Using the dq.addLast("")
 - Adding elements on the left side

```
// Java program to demonstrate the
// addition of elements in deque

import java.util.*;

public class ArrayDequeDemo {
    public static void main(String[] args)
    {
        // Initializing an deque
        Deque<String> dq
            = new ArrayDeque<String>();

        // add() method to insert
        dq.add("For");
        dq.addFirst("Geeks");
        dq.addLast("Geeks");

        System.out.println(dq);
    }
}
```

Output:

```
[Geeks, For, Geeks]
```

- Remove the elements from either end
 - *removeFirst()*, *removeLast()* commands
- Deque Pop
 - Remove the head element from the deque by using the *Deque.pop()* method and display the popped element. In the output

```
16 System.out.println("Element popped: "+d.pop());
17 System.out.println("Element popped: "+d.pop());
18
19 System.out.println("Elements in the deque are: "+d);
20
21 }
22 }
```

- Deque Pull
 - used to retrieve or fetch and remove the element present at the head of the Deque.

- What is your data structure useful for?

- Where would you want to use it, vs where is it not useful?
 - Algorithms in program
 - Any time operations are different between the front end or rear end at a line
 - Use Wal-Mart queue for automotive maintenance
 - remove or add elements to the ends of a linear sequence efficiently
 - Other examples: undo lists, Trash emptying at end of 30-days

Implementation:

- What implementation of the data structure did you get?
 - ListedArray
- How did you implement it?
 - Removing and adding elements from both sides
 - Using push and pull
- What are the most basic methods needed to access/store/and modify data in your structure
 - Nodes
 - Add
 - Variables
 - Pop
 - Remove
 - DisplayTheList
- How do you go about adding or deleting elements?

Not sure if this diagram will be of any use. Placing here in case,

