```
package priorityqueueusingarray;

import java.util.Arrays;

class PriorityQueue {

    private int[] items = new int[5];
    private int count;

    // O(n)
    public void add(int item) {
        if (isFull()) {
            throw new IllegalStateException();
        }

        var i = shiftItemsToInsert(item);
        items[i] = item;
        count++;
    }

    public boolean isFull() {
        return count == items.length;
    }

    private int shiftItemsToInsert(int item) {
        int i;
        for (i = count - 1; i >= 0; i--) {
            if (items[i] > item) {
                items[i + 1] = items[i];
            } else {
                break;
            }
        }
        return i + 1;
    }

    // O(1)
    public int remove() {
        if (isEmpty()) {
            throw new IllegalStateException();
        }

        return items[--count];
    }

    public boolean isEmpty() {
        return count == 0;
    }

    @Override
    public String toString() {
        return Arrays.toString(items);
    }
}
public class PriorityQueueUsingArray {
```

```
    public static void main(String[] args) {
        PriorityQueue queue = new PriorityQueue();
        queue.add(5);
        queue.add(3);
        queue.add(6);
        queue.add(1);
        queue.add(4);
        System.out.println(queue);
    }

}
```