FACE
Prep

Sort without extra space

FACE Prep

Sort the queue

Problem:

Sort the given queue without using any extra space

FACE Prep

Queue

Queue follows FIFO i.e) First In First Out

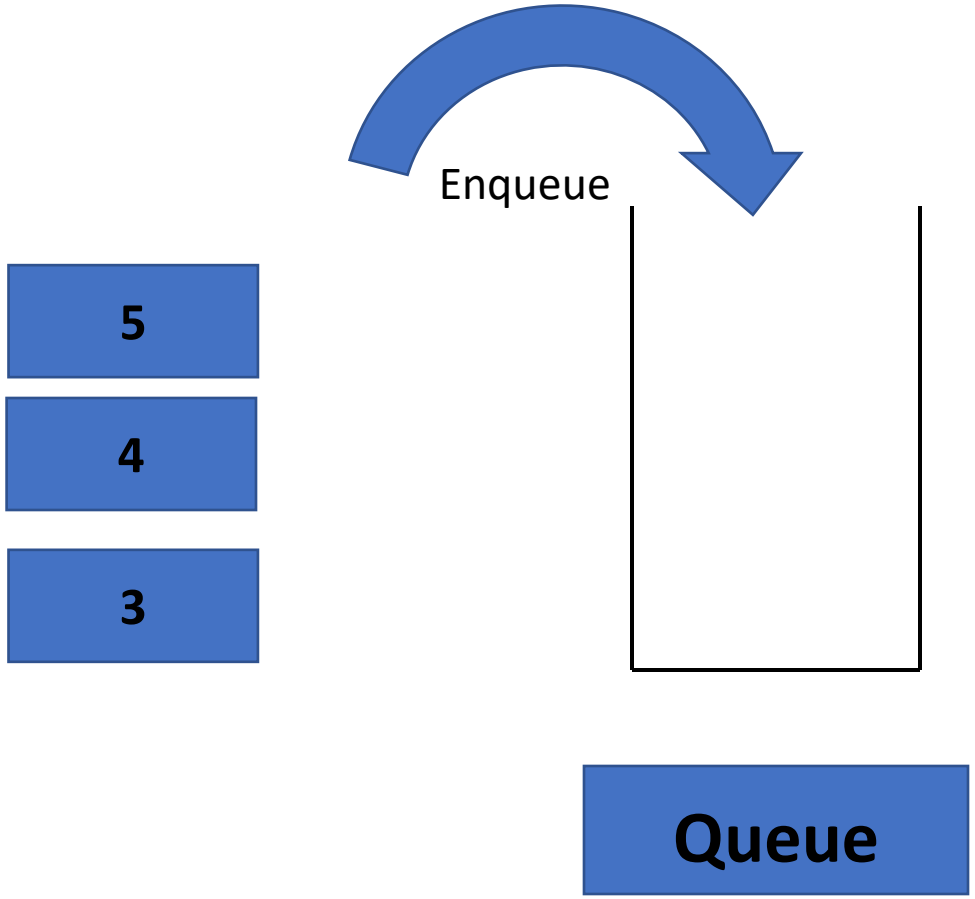
Insertion is said to be enqueue, which can be done in rear end

Deletion is said to be dequeue, which can be done in front end

FACE Prep

Queue

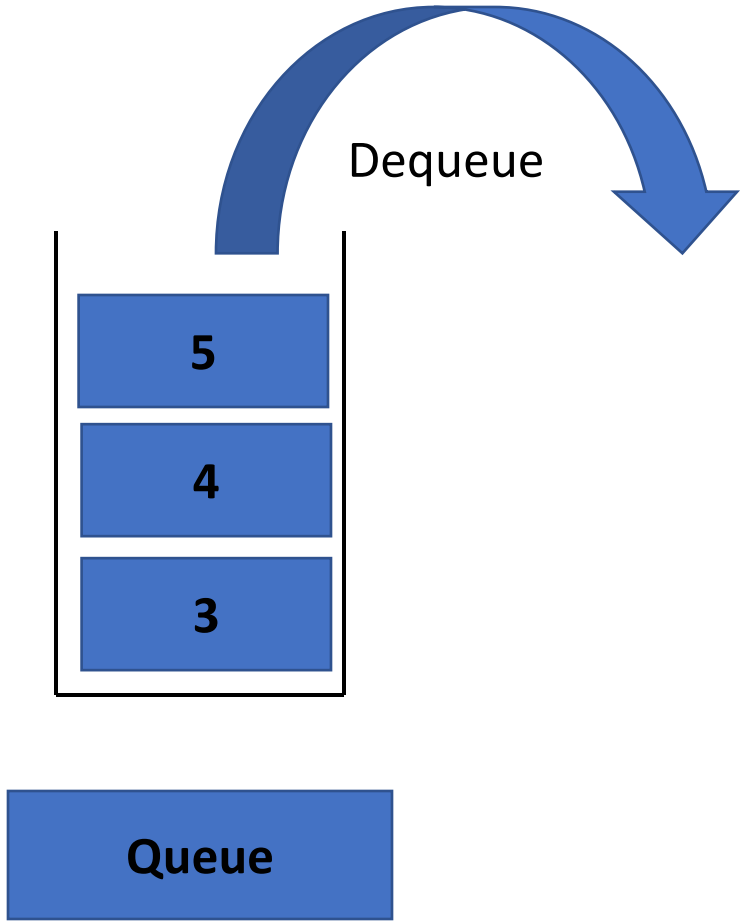
Enqueue – First In



FACE Prep

Queue

Dequeue – First Out



FACE Prep

FACE Prep

```
1   import java.util.*;
2   class Main
3   {
4         public static int minIndex(Queue<Integer> list, int sortIndex)
5         {
6         int min_index = -1;
7         int min_value = Integer.MAX_VALUE;
8         int s = list.size();
9         for (int i = 0; i < s; i++)
10        {
11              int current = list.peek();
12              list.poll();
13              if (current <= min_value && i <= sortIndex)
14              {
15                    min_index = i;
16                    min_value = current;
17              }
18              list.add(current);
19        }
20        return min_index;
21  }
22
```

FACE Prep

```
23 public static void insertMinToRear(Queue<Integer> list, int min_index) {
24             int min_value = 0;
25             int s = list.size();
26             for (int i = 0; i < s; i++)
27             {
28             int current = list.peek();
29             list.poll();
30             if (i != min_index)
31                   list.add(current);
32             else
33                   min_value = current;
34             }
35             list.add(min_value);
36       }
37       public static void sortQueue(Queue<Integer> list) {
38             for(int i = 1; i <= list.size(); i++)
39             {
40                   int min_index = minIndex(list,list.size() - i);
41                   insertMinToRear(list, min_index);
42             }
43       }
44
```

FACE Prep

```
45 public static void main (String[] args)
46       {
47             Queue<Integer> list = new LinkedList<Integer>();
48             list.add(6);
49             list.add(11);
50             list.add(15);
51             list.add(4);
52             sortQueue(list);
53             while(list.isEmpty()== false)
54             {
55                   System.out.print(list.peek() + " ");
56                   list.poll();
57             }
58       }
59 }
60
61
62
63
64
65
66
```

FACE Prep

THANK YOU

FACE Prep