

Code

Main.java

```
1 public class Main {
2     public static void main(String[] args) {
3         // Create a new MinQueue object of Integer type
4         MinQueue<Integer> minQueue = new MinQueue<>();
5
6         // Add some elements to the queue
7         minQueue.add(5);
8         minQueue.add(3);
9         minQueue.add(7);
10        minQueue.add(2);
11
12        // Print the current minimum element and size of the queue
13        System.out.println("Current minimum element: " + minQueue.min());
14        System.out.println("Size of the queue: " + minQueue.size());
15
16        // Remove an element from the queue
17        minQueue.remove();
18
19        // Print the updated minimum element and size of the queue
20        System.out.println("Current minimum element: " + minQueue.min());
21        System.out.println("Size of the queue: " + minQueue.size());
22    }
23 }
```

```
1  import java.util.Deque;
2  import java.util.LinkedList;
3  import java.util.Queue;
4
5  public class MinQueue<T extends Comparable<T>> {
6      private Queue<T> queue; // Queue to store elements
7      private Deque<T> minDeque; // Deque to keep track of minimum element
8
9      // Constructor to initialize the queue and deque
10     public MinQueue() {
11         queue = new LinkedList<>();
12         minDeque = new LinkedList<>();
13     }
14
15     // Method to add an element to the queue
16     public void add(T x) {
17         queue.add(x); // Add element to queue
18
19         // Remove all elements from the back of the deque that are larger
20         // than the new element
21         while (!minDeque.isEmpty() && minDeque.peekLast().compareTo(x) >
22             0) {
23             minDeque.removeLast();
24         }
25         // Add the new element to the back of the deque
26         minDeque.addLast(x);
27     }
28 }
```

```

25     }
26
27     // Method to remove the first element from the queue
28     public T remove() {
29         T x = queue.remove(); // Remove element from queue
30
31         // If the removed element was the minimum element, remove it from
the front of the deque
32         if (x.equals(minDeque.peekFirst())) {
33             minDeque.removeFirst();
34         }
35         return x;
36     }
37
38     // Method to return the minimum element in the queue
39     public T min() {
40         return minDeque.peekFirst(); // Return the element at the front
of the deque, which is the minimum element
41     }
42
43     // Method to return the size of the queue
44     public int size() {
45         return queue.size();
46     }
47

```

Pseudo Code

Class MinQueue<T extends Comparable<T>>:

Node head, tail

Node minNode

Function add(x: T):

newNode = new Node(x)

if tail is not null:

tail.next = newNode

```
    newNode.prev = tail  
  
tail = newNode  
  
if head is null:  
  
    head = tail  
  
if minNode is null or x < minNode.value:  
  
    minNode = newNode
```

Function remove():

```
if head is null:  
  
    throw new NoSuchElementException()  
  
removedNode = head  
  
head = head.next  
  
if head is null:  
  
    tail = null  
  
else:  
  
    head.prev = null  
  
if removedNode == minNode:  
  
    updateMinNode()  
  
return removedNode.value
```

Function size():

```
count = 0  
  
node = head  
  
while node is not null:
```

```
count++  
  
node = node.next  
  
return count
```

Function min():

```
if minNode is null:  
  
    throw new NoSuchElementException()  
  
return minNode.value
```

Function updateMinNode():

```
if head is null:  
  
    minNode = null  
  
else:  
  
    node = head  
  
    minNode = head  
  
    while node is not null:  
  
        if node.value < minNode.value:  
  
            minNode = node  
  
        node = node.next
```

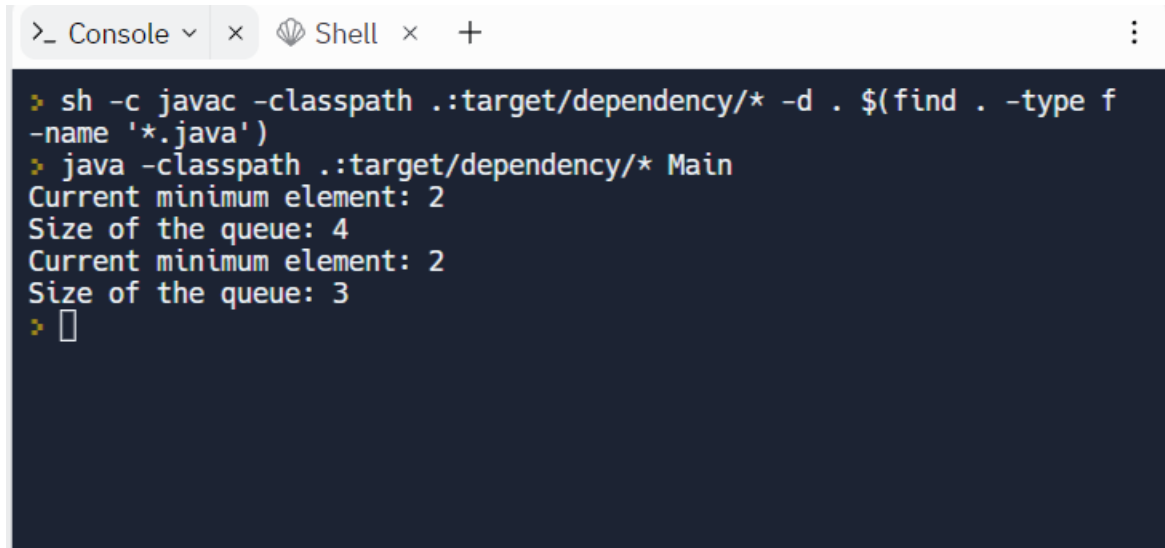
Class Node:

```
T value  
  
Node prev, next
```

Constructor Node(x: T):

value = x

Outputs



```
>_ Console x Shell x +  
➤ sh -c javac -classpath .:target/dependency/* -d . $(find . -type f  
-name '*.java')  
➤ java -classpath .:target/dependency/* Main  
Current minimum element: 2  
Size of the queue: 4  
Current minimum element: 2  
Size of the queue: 3  
➤ □
```

Figure 1 Output Screen

REPL.IT LINK

<https://replit.com/@ravipanchal3/UnacceptableExpensiveOpengl#MinQueue.java>