



```
public static void reverse(Queue<Integer> Q) {  
    if (Q.isEmpty()) {  
        return;  
    }  
    Integer data = Q.remove();  
    reverse(Q);  
    Q.add(data);  
}
```

The diagram illustrates the control flow of the provided Java code. It features several orange circular nodes connected by arrows. One node is positioned at the start of the first line of code. Another node is located on the line containing the `if` statement. A third node is placed on the `return;` line. A fourth node is situated on the line containing the closing curly brace of the `if` block. A fifth node is located on the line containing the recursive call `reverse(Q);`. A sixth node is positioned on the line containing `Q.add(data);`. Arrows indicate the flow of execution: from the start node to the `if` node, from the `if` node to the `return;` node, from the `return;` node to the closing brace node, from the closing brace node to the recursive call node, and from the recursive call node to the `Q.add(data);` node. There is also a direct arrow from the start node to the recursive call node, representing the recursive jump.