**Day6: Task3**

**You have a queue of integers that you need to sort. You can only use additional space equivalent to one stack. Describe the steps you would take to sort the elements in the queue.**

sort the elements in the queue using only one stack

**1. \*Enqueueing Elements\*:**

- Begin by enqueueing all elements from the original queue into the stack.

**2. \*Sorting in Stack\*:**

- While the stack is not empty, pop an element from the stack.

- If the popped element is greater than or equal to the front element of the queue, enqueue it into the queue.

- Otherwise, while the front element of the queue is greater than the popped element, dequeue an element from the queue and push it into the stack.

- Once the popped element is successfully inserted into the queue in the correct position, repeat the process until the stack is empty.

**3. \*Dequeueing Elements\*:**

- Now, all elements are in the queue in ascending order.

- Dequeue all elements from the queue and enqueue them back into the original queue.

**4. \*Result\*:**

- After completing the above steps, the original queue will contain the sorted elements.

This approach effectively sorts the elements in the queue using limited additional space equivalent to one stack.

To sort the elements in a queue using only one additional stack, you can utilize a variation of the merge sort algorithm. Here's a step-by-step approach:

1. **Divide and Conquer using Recursion:**

Divide the queue into two halves recursively until each half contains only one element. You can achieve this by transferring elements from the queue to the stack until the queue is empty, then transferring half of the elements back to the queue.

Repeat the following steps recursively until the queue contains only one element:

* Pop elements from the front of the queue and push them onto the stack until the queue is empty.
* Pop half of the elements from the stack and push them back onto the queue.

1. **Merge the Halves:**

After the recursion, you will have two sorted halves in the queue. Merge these two halves back into a single sorted queue. You can do this by comparing elements from the fronts of the two halves and enqueueing the smaller one into a temporary stack. Once one of the halves is empty, transfer the remaining elements from the non-empty half to the temporary stack. Then, transfer the elements from the temporary stack back to the original queue.

* Create a temporary stack.
* While both the queue and the stack are not empty, compare the front elements of both.
  + - * 1. If the front element of the queue is smaller, dequeue it and push it onto the temporary stack.
        2. If the front element of the stack is smaller, pop it from the stack and enqueue it back onto the queue.
* Once one of the halves is empty, transfer the remaining elements from the non-empty half to the temporary stack.
* Transfer the elements from the temporary stack back to the original queue.