```
from linked_queue import LinkedQueue

def merge(S1, S2, S):
  """Merge two sorted queue instances S1 and S2 into empty queue S."""
  while not S1.is_empty() and not S2.is_empty():
    if S1.first() < S2.first():
      S.enqueue(S1.dequeue())
    else:
      S.enqueue(S2.dequeue())
  while not S1.is_empty():            # move remaining elements of S1 to S
    S.enqueue(S1.dequeue())
  while not S2.is_empty():            # move remaining elements of S2 to S
    S.enqueue(S2.dequeue())

def merge_sort(S):
  """Sort the elements of queue S using the merge-sort algorithm."""
  n = len(S)
  if n < 2:
    return                            # list is already sorted

  # divide

  S1 = LinkedQueue()                  # or any other queue implementation
  S2 = LinkedQueue()
  while len(S1) < n // 2:             # move the first n//2 elements to S1
    S1.enqueue(S.dequeue())
  while not S.is_empty():             # move the rest to S2
    S2.enqueue(S.dequeue())

  # conquer (with recursion)

  merge_sort(S1)                      # sort first half
  merge_sort(S2)                      # sort second half

  # merge results

  merge(S1, S2, S)                    # merge sorted halves back into S
```