```
from linked_queue import LinkedQueue

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

  # divide

  p = S.first()                         # using first as arbitrary pivot
  L = LinkedQueue()
  E = LinkedQueue()
  G = LinkedQueue()
  while not S.is_empty():               # divide S into L, E, and G
    if S.first() < p:
      L.enqueue(S.dequeue())
    elif p < S.first():
      G.enqueue(S.dequeue())
    else:                               # S.first() must equal pivot
      E.enqueue(S.dequeue())

  # conquer (with recursion)

  quick_sort(L)                         # sort elements less than p
  quick_sort(G)                         # sort elements greater than p

  # concatenate results

  while not L.is_empty():
    S.enqueue(L.dequeue())
  while not E.is_empty():
    S.enqueue(E.dequeue())
  while not G.is_empty():
    S.enqueue(G.dequeue())
```