```
from merge_array import merge_sort

class _Item:
  """Lightweight composite to store decorated value for sorting."""
  __slots__ = '_key', '_value'

  def __init__(self, k, v):
    self._key = k
    self._value = v

  def __lt__(self, other):
    return self._key < other._key    # compare items based on their keys

def decorated_merge_sort(data, key=None):
  """Demonstration of the decorate-sort-undecorate pattern."""
  if key is not None:
    for j in range(len(data)):
      data[j] = _Item(key(data[j]), data[j])          # decorate each element
  merge_sort(data)                                    # sort with existing
algorithm
  if key is not None:
    for j in range(len(data)):
      data[j] = data[j]._value                        # undecorate each element
```