```
def merge_sort(orders):
    if len(orders) <= 1:
        return orders

    mid = len(orders) // 2
    left_half = orders[:mid]
    right_half = orders[mid:]

    left_sorted = merge_sort(left_half)
    right_sorted = merge_sort(right_half)

    return merge(left_sorted, right_sorted)

def merge(left_half, right_half):
    merged_list = []
    i = 0
    j = 0

    while i < len(left_half) and j < len(right_half):
        if left_half[i]['delivery_time'] < right_half[j]['delivery_time']:
            merged_list.append(left_half[i])
            i += 1
        else:
            merged_list.append(right_half[j])
            j += 1

    while i < len(left_half):
        merged_list.append(left_half[i])
        i += 1

    while j < len(right_half):
        merged_list.append(right_half[j])
        j += 1

    return merged_list

if __name__ == "__main__":
    online_orders = [
        {'order_id': 'A1', 'delivery_time': 55},
        {'order_id': 'B2', 'delivery_time': 25},
        {'order_id': 'C3', 'delivery_time': 60},
```

```
    {'order_id': 'D4', 'delivery_time': 20},
    {'order_id': 'E5', 'delivery_time': 40},
    {'order_id': 'F6', 'delivery_time': 30},
    {'order_id': 'G7', 'delivery_time': 50},
]

print("Original list order:")
for order in online_orders:
    print(f"Order {order['order_id']} -> Delivery Time: {order['delivery_time']} minutes")

sorted_orders = merge_sort(online_orders)

print("\nSorted list of orders by delivery time:")
for order in sorted_orders:
    print(f"Order {order['order_id']} -> Delivery Time: {order['delivery_time']} minutes")
```