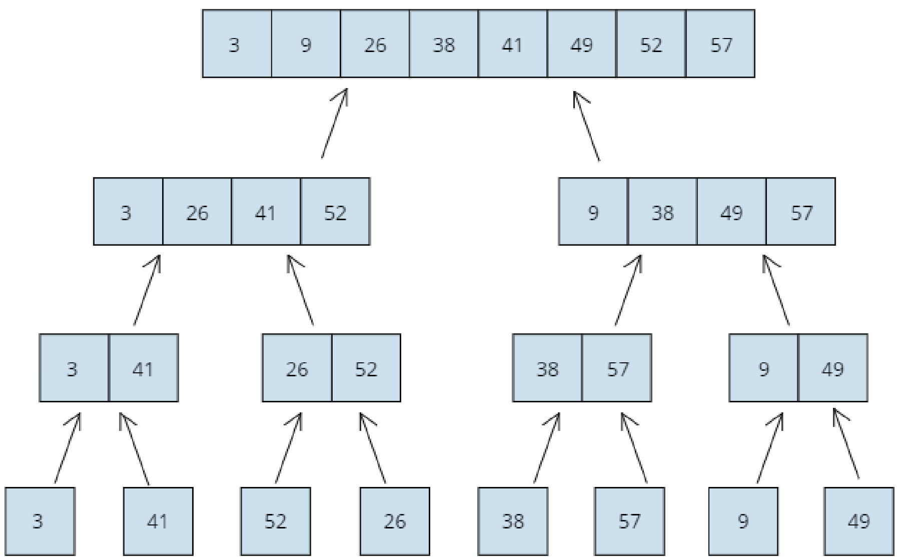
1 Using Figure 2.4 as a model, illustrate the operation of merge sort on the array A=(3,41,52,26,38,57,9,49).



2 Rewrite the Merge grocedure so that it does not use sentinels? instead stopping pnce either array L or Rhas had all elements copied back to A and then colying the remainder of the other array back into A

```
def _merge(A,p,r,q):
```

```
    L = A[p:r+1];
    R = A[r+1:q+1];
    index = p;
    while len(R)>0 and len(L)>0:
        if (L[0]<R[0]):
            A[index] = L.pop(0);
        else:
            A[index] = R.pop(0);
        index = index+1;
    for i in range(0, len(L),1):
        A[index] = L.pop(0);
        index = index + 1;
    for i in range(0, len(R), 1):
        A[index] = R.pop(0);
        index = index + 1;

def _sort(array, l, r):
    if r >l:
        _sort(array,l, int((l+r)/2));
        _sort(array, int((l+r)/2+1),r);
        _merge(array, l, int((l+r)/2),r)

def sort(array):
    _sort(array,0, len(array)-1);
    return array;

print(sort([1,2,3,4,5]));
print(sort([5,4,3,2,1]));
print(sort([5,4,1,2,3]));
print(sort([1]));
print(sort([]));
```

3 2.3-3

Use mathematical induction to show that when n is an exact power of 2, the solution of the recurrence

$$T(n) = \begin{cases} 2, & \text{if } n = 2 \\ 2T(n/2) + n, & \text{if } n = 2^k\text{, for } k > 1 \end{cases}$$

is $T(n) = nlg(n)$

For $n = 2$: $T(2) = 2 = 2lg(2) = nlg(n)$

For $n = 4$: $T(4) = 2T(2) + 4 = 2 * 2lg(2) + 4 * 1 = 4 * lg(2) + 4 * lg(2) = 4 * lg(4) = nlg(n)$

Assume, that for all $n \leq k$ T(n)=nlg(n). For $n = k * 2$: $T(k * 2) = 2T(k * 2/2) + 2k = 2k * lg(k) + 2k = 2k * lg(k) + 2k * lg2 = 2k * (lgk + lg2) = 2k * lg(2k)$

4 2.3-4

We can express insertion sort as a recursive procedure as follows. In order to sort A[1..n], we recursively sort A[1..n-1] and than insert A[n] into the sorted array A[1..n-1]. Write a recurrence for the worst-case running time of this recursive version of insertion sort.

$$T(n) = \begin{cases} \Theta(1), & \text{if } n = 1 \\ T(n-1) + n - 1, & \text{if } n > 1 \end{cases}$$